Deep Learning for Genome-Wide Association Studies and the Impact of SNP Locations

by

© *Songyuan Ji*

A thesis submitted to the

School of Graduate Studies

in partial fulfilment of the

requirements for the degree of

Master of *Science*

Department of *Computer Science*

Memorial University of Newfoundland

September, 2019

St. John's Newfoundland

Abstract

The study of Single Nucleotide Polymorphisms (SNPs) associated with human diseases is important for identifying pathogenic genetic variants and illuminating the genetic architecture of complex diseases. A Genome-wide association study (GWAS) examines genetic variation in different individuals and detects disease related SNPs. The traditional machine learning methods always use SNPs data as a sequence to analyze and process and thus may overlook the complex interacting relationships among multiple genetic factors. In this thesis, we propose a new hybrid deep learning approach to identify susceptibility SNPs associated with colorectal cancer. A set of SNPs variants were first selected by a hybrid feature selection algorithm, and then organized as 3D images using a selection of space-filling curve models. A multi-layer deep Convolutional Neural Network was constructed and trained using those images. We found that images generated using the space-filling curve model that preserve the original SNP locations in the genome yield the best classification performance. We also report a set of high risk SNPs associate with colorectal cancer as the result of the deep neural network model.

Acknowledgements

I would like to express my sincere thanks to my supervisors, Dr. Minglun Gong, Dr. Ting Hu and Dr. Yuanzhu Chen for providing me with the research project, Deep Learning for Bioinformatics. My detailed discussions with them and their encouragement of innovative ideas and critical thinking were critical for guiding me to be an independent researcher.

I would like to acknowledge the financial supports from the Computer Vision lab, Department of Computer Science, School of Medicine and School of Graduate Study.

Additionally, I would like to give my special thanks to the Dr. Zili Yi and Mr. Shiyao Wang for detailed discussions. They provided professional suggestions of programming, scientific computing and math derivations. Finally, I also appreciate all the reviews and suggestions for from the examiner.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Genome-wide association studies

Single nucleotide polymorphisms (SNPs) are DNA sequence variations that occur in a single nucleotide in the genome of chromosomes. SNPs have the characteristics of breadth, representation, heredity and stability.Because of the breadth property, SNPs are widespread in the human genome and there are estimated to be more than three million totally. Some SNPs in genes may directly affect the protein structure or expression level, so they may represent some acting factors in the genetic mechanism of diseases. Moreover, the heredity property refers to a species polymorphism caused by a single nucleotide (ATGC) mutation in a genomic DNA sequence. Finally, the stability characteristics refers to self-crossing of parents. There will be no separation of traits, and the parents' traits will be passed on to their offspring compared with repeat sequence polymorphism markers. SNPs have higher genetic stability [49]. SNPs are the most abundant genetic variation in the human genome [66]. The SNPs

in coding region are likely to affect protein structure, expression levels [70], and the genetic mechanisms of the disease that is associated with them. Thus, SNPs association research is significant for locating pathogenic genes and discovering the genetic mechanisms of complex diseases [32, 80].

The colorectal cancer (CRC) is a high prevalence cancer, and its rate is 5.2% for men and 4.8% for women according to statistics in the United States [71]. There is some heritability that remains elusively although several genome-wide association studies (GWAS) of CRC have successfully identified common SNPs associated with CRC [71]. In recent years, the rapid development of gene chip technology has made high-throughput detection of SNPs data simpler and less expensive [80]. As the number of cases tested in the experiment can't be increased more much in the official database from medical institutions, we have to improve the existing algorithms to get more useful SNPs [61]. As a result, the number of samples in SNPs dataset is much smaller than the number of SNPs [61]. Given the high dimensionality and small sample size of SNPs data, how to effectively execute mass data association analysis becomes the first major difficulty in SNPs association research [61, 66]. GWAS is also known as a processing that refers to find the SNPs mutation sequence related to disease in whole human genome [71], and its work principle is shown in the following Figure 1.1.

The complexity of the genetic disease pathology is another issue of GWAS. Recent research shows that complex diseases have more complicated etiology, usually involving the combination of multiple genes [49]. However, the inheritance of SNPs is characterized by linkage disequilibrium, that is, there is a linkage relationship between alleles at different genome location, rather than free association [50]. This

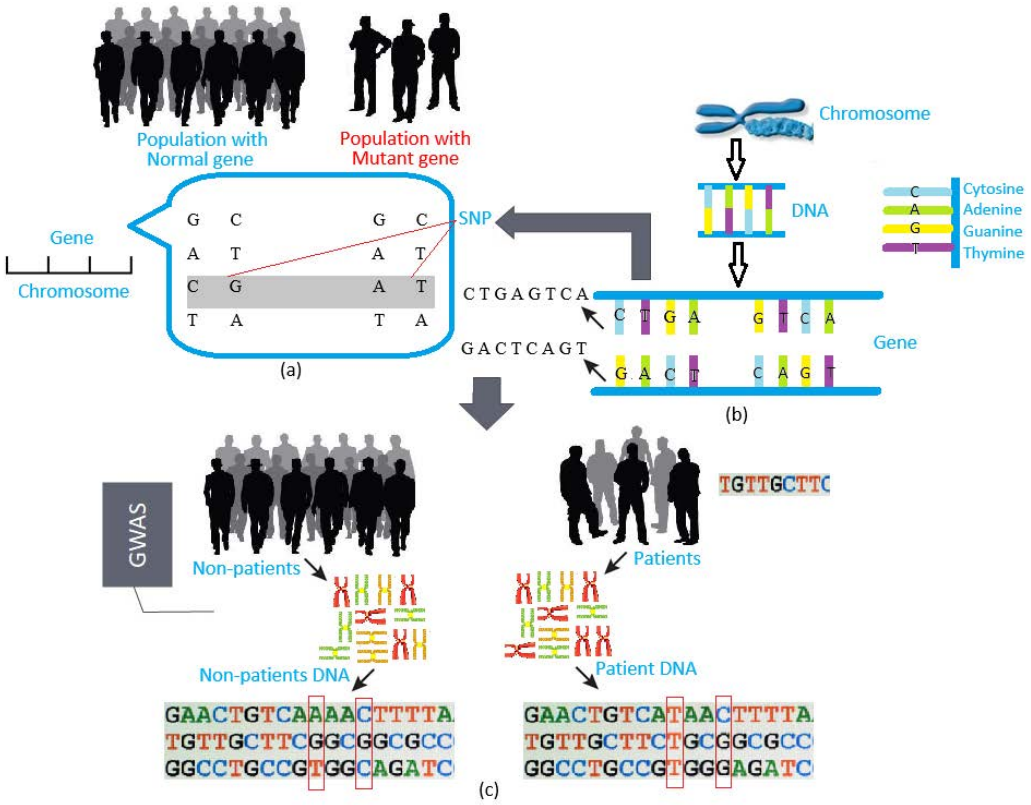

Figure 1.1: Genome-Wide Association Study [22]. (a) The difference between the normal gene and mutant gene. (b) The location of SNP in the chromosome. (c) The gene in patients and healthy individuals.

research does not delve into the pathogenesis of SNPs from a biomedical perspective but takes the analysis for SNPs from feature selection combined with deep learning, and this research considers the effects between SNPs rather than the impact of a single SNP [54]. In recent years, SNPs research has developed very quickly. In the field of bioinformatics, many researchers use existing statistical analysis tools, such as Weka, R language packages, PLINK, and other SNPs data analysis tools [61, 71]. Many existing statistical analysis tools depend on the correlation between individual

SNPs and diseases as the basis SNPs to filter, ignoring the interaction between SNPs, which led to the limited heritability that can be explained by the filtrate results [30].

1.2 Feature selection and classification for GWAS

Genome-wide Association Analysis(GWAS) aims to find SNPs in genome-wide ranges that have the highest degree of association with a disease [71]. Various research methods are applied to the research of SNPs, and these methods consider the relationship between SNPs, such as data mining, machine learning and pattern recognition. However, research of SNPs is costly, and its computing cost is significant [49]. Finding new analysis methods that can effectively reduce the dimension as well as fully consider the interaction among SNPs is necessary.

Feature Selection is the one of major concerns for GWAS. GWAS data often includes up to a million SNPs. Thus, an effective scheme for feature selection needs to be designed to identify the most relevant SNPs. Feature selection algorithms are divided into two types based on the evaluation strategy of the feature set. Filter algorithms are often efficient but are independent of the subsequent classification training [35]. Wrapper algorithms are often part of the classifier but are computational demanding as a result of the iterative training process [8]. In this thesis, we combined a filter-style ReliefF algorithm and a wrapper-style Genetic Algorithm (GA).

Data visualization is the presentation of data in a pictorial or graphical format. It enables decision-makers to see the structure of and relationship of attributes in order to grasp difficult concepts or identify new patterns [4]. Space filling curves can be used to transform a sequence data into a 2D image. The range of the curve contains

the entire 2D unit square (or more generally the n-dimensional unit hypercube) [33]. A continuous curve of 2 or 3 (or higher) dimensions can be intuitively considered as a path of continuous moving points [37]. The curve is a continuous function whose domain is the unit interval [0,1] [29]. In the most general form, the scope of such a function can be in any topological space. However, in most studies, the range will only be defined in the Euclidean space, such as a 2D plane (planar curve) or a 3D space (space curve) [29]. If curves have no endpoints, they are generally defined as continuous function [41].

Deep Learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations as opposed to task-specific algorithms [15, 46]. The learning can be supervised, semi-supervised or unsupervised. Some representations are loosely based on interpretation of information processing and communication patterns in a biological nervous system [52], such as neural coding that attempts to define a relationship between various stimuli and associated neuron responses in the brain [53]. Deep Learning architectures, such as deep neural networks, deep belief networks and recurrent neural networks, have been applied to computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics and drug design [55]. Deep Learning has produced results equivalent to human experts, and in some cases, better than human experts. Deep Learning technologies can be used to search (collection and screening), process (edit, organize, manage and display) and analyze (calculation and simulation) biological data.

1.3 Thesis contributions

In this thesis, we employ a hybrid feature selection method using both the ReliefF and Genetic Algorithms, and subsequently use CNN to classify the images transformed from GWAS data (i.e., SNPs sequences). Our method not only considers SNP's values but also incorporates the distance information of SNPs on DNA. By studying a colorectal cancer GWAS dataset collected from the Newfoundland population, we found a subset of SNPs that can be used to predict the disease status with accurately. Also, we test the performance of hybrid feature selection method to verify the model in this research, and investigate different space-filling curves. Our results suggest that transforming the sequential SNPs data into two-dimensional (2D) images improves the prediction and the best result is achieved by using space-filling curves that preserve the location information of SNPs on the DNA sequence.

Chapter 2

Background

2.1 GWAS and data preprocessing

Quality control (QC) is a critical element in GWAS. Since millions of genotypes are generated, even a small percentage of genotyping error can lead to spurious GWAS results. QC can be considered to have two aspects: genotyping chips (i.e. issues related to making genotype calls from intensity measurements) and downstream QC issues. In this thesis, we focus on downstream QC approaches, i.e. data cleaning procedures that can be applied once we already have genotype calls. Downstream QC covers two major areas of quality: subject-based quality measures and variant-based quality measures. The specific QC measures for the two domains are as follows.

1. **Subject-Based Measures**: Subject-Based Measures Rate is the proportion of missing genotypes per subject. The Gender parameter refers to check that self-reported gender matches genotyped gender. Relatedness is undisclosed familial relationships and duplicate enrollment. Replicate Discordance is the agreement

with independent genotyping. Population Outliers refers to the subjects with significantly different genetic background from the rest of study samples.

2. Variant-Based Measures: Variant-Specific Missingness Rate is the proportion of failed assays for a variant. Minor Allele Frequency refers to very low-frequency alleles are more likely to represent genotyping error and can give spurious association results. Hardy-Weinberg equilibrium is a principle stating that the genetic variation in a population will remain constant from one generation to the next in the absence of disturbing factors [71]. Mendelian Errors means family data evidence of non-Mendelian transmission. Replicate Discordance refers to replicating and keeping SNPs with genotype discordance rate as requirement (The genotype discordance rate is the number of genotype calls that differ between a pair of samples divided by the total number of SNPs for which both calls are non-missing) [43].

2.2 Feature selection

High-dimensional data mining is challenging, especially in text mining, image processing and data analysis. Therefore, dimensionality reduction is a necessary task in the process of data pre-treatment. There are two broad categories of dimensionality reduction. One is to extract new features from the original ones, and another is to select a subset from the original features, which is called feature selection or best subset selection. The goal of feature selection is to select the most relevant and effective features. Using feature selection enables the machine learning algorithm to train faster, reduces the complexity of a model and makes it easier to interpret, improves

the accuracy of a model if the right subset is chosen, and reduces over-fitting [31]. Generally, there are three feature selection methods.

1. Filter feature selection methods are generally used as a preprocessing step. The main idea is to "rate" the characteristics of each feature, that is, to assign weights to each feature which represent the importance of that feature. The major algorithms include Relief, ReliefF and Information Gain [19].

2. Regarding to wrapper feature selection methods, we try to use a subset of features and train a model using them. Based on the inferences that we draw from the previous model, we decide to add or remove features from this subset. The problem is essentially reduced to a search problem [8]. These methods are usually computationally expensive, and can be solved by optimization algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) [8].

3. Embedded feature selection methods combine the features of the filter and wrapper methods, implemented by an algorithm with a built-in feature selection method. The main idea is to learn the best attributes to improve the accuracy of the model in the model environment [82]. In the process of determining the model, select the features that are beneficial to model training. The ridge regression method is an example of this method type, which has inbuilt penalization functions to reduce over-fitting [82].

2.3 Data visualization

Data visualization is the study of the visual representation of data. The visual representation of data is a type of information extracted in a summary form, including various attributes of each corresponding information individual [24]. Using charts or images to summarize complex data ensures that relationships are understood faster than reports or spreadsheets [79]. Furthermore, data visualization encourages users to explore and even manipulate data to discover new patterns and knowledge.

2.3.1 Data visualization and applications for bioinformatics

Data visualization has become the general tool for modern business intelligence (BI), artificial intelligence (AI), engineering and bioscience [14]. In the research field of bioinformatics, there are many applications and methods of data visualization such as the Treemap, Sunburst Diagram, Stream Graph, Circos, and space-filling curve [28]. Treemap is suitable for presenting data with hierarchical relationships, which can visually reflect the comparison between peers [28]. The Sunburst Diagram is a modern pie chart that transcends traditional pie charts and ring charts, expresses clear levels and attributions, and displays data composition in a parent-child hierarchy [28]. The Stream Graph uses a flowing organic shape, which allows it to show the variation of different categories of data over time [28]. Among them, the space-filling curve is a relatively new method of data visualization in bioinformatics and is most suitable for the purpose of this thesis.

2.3.2 Space-filling curve

In mathematical analysis, a space-filling curve (SFC) is a curve whose range contains the entire 2D unit square (or more generally an n-dimensional unit hypercube). Sometimes, the curve is identified with the range or image of the function (the set of all possible values of the function) [44].

The space-filling curve is an approximate representation method. The data space is divided into grids of the same size, and these grids are coded according to certain methods. A spatial object consists of a set of grids, each of which specifies a unique encoding and maintains spatial proximity to some extent, and the labels of adjacent grids are also adjacent to each other. In this way, multidimensional spatial data can be dimensionally reduced into a one dimensional (1D) space [2]. Correspondingly, SFC can also convert 1D data into a multidimensional space [33]. Space-filling curves have been widely used as mappings from 1D spaces to multi-dimensional spaces. The method of mapping 1D domain to multidimensional space plays an important role in many fields. The advantage of using spatial fill curves to define a multidimensional space is that it not only maintains the value of each element in a 1D sequence, but also presents a nonlinear relationship between elements [44]. At the same time, search, scheduling, space access, indexing and clustering operations can be implemented [2].

In this thesis, mapping provides pre-processing steps for multidimensional data applications (feature selection). Preprocessing takes an 1D data (sequence) as input and outputs it as a multidimensional data (matrix). This idea keeps the existing feature selection algorithms and data structures independent of the dimensionality of data [2]. The purpose of the mapping is that each data value is the original 1D array

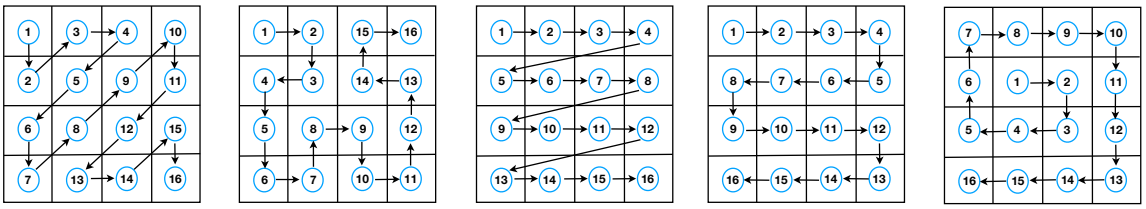

Figure 2.1: Various space-filling curves [44]. (a), (b), (c), (d) and (e) are images of various space-filling curves, and the scanning path of all elements in a sequence (size is $2^{rank} \times 2^{rank}$, $rank = 2$). From 1 to 16 are the elements in sequence.

A Cantor curve is a metrizable 1D continuum [77]. In mathematics, a nowhere dense continuum on a topological space is a closed uninterrupted set [2], and it is the first characterization of 1D closed connected subsets of the plane [77]. A Cantor curve contains a nowhere-dense subcontinuum if and only if the closure of the set of its branching points is 1D [77]. The scanning method of a Cantor curve is to order the grid points in alternating directions along parallel diagonals of the grid. The scan pattern is affected by the direction of the first step. Therefore, the Cantor curve is also called zigzag scanning [12]. The Cantor curve is shown in Figure 2.1(a).

A Hilbert curve is a continuous and unguided curve [29]. It can linearly run through each discrete unit of two or more dimensions and pass only once, and linearly sort and encode each discrete unit as a unique identifier for the unit. Also, a Hilbert curve is a curve without any intersections and overlaps. Two continuous functions

$x = f(t)$ and $y = g(t)$ can be defined, where the function f is to calculate the abscissa x of the point, and the function g is to calculate the ordinate y of the point. When the parameter t is in the interval of 0, 1, the set of points

$$\{(x_i, y_i) \in square, i = 1, 2, 3, ...2^n * 2^n\}$$

makes it possible to fill a flat square. The Hilbert Curve is shown in Figure 2.1(b).

A Row curve is a general curve. The idea of this curve refers uses the top left grid cells [0,0] in the matrix as the first point of the curve, then start accessing the data from left to right in each row until the end of the bottom right grid cell $[n, n]$ [65]. The Row curve is shown in Figure 2.1(c).

A Row-prime curve is a unique curve which is a variant of a Row curve. Its accessing rule is to read the line from left to right and then from right to left, which is called Row prime sort [65]. The Row-prime curve is shown in Figure 2.1(d).

The basic idea of spiral curve is that a point is first put in the center pixel of the image, and continuously moved into adjacent pixels following a clockwise spiral path outward [33]. The Spiral curve is shown in Figure 2.1(e).

2.4 Deep learning for bioinformatics

Deep learning is a branch of machine learning (ML) in artificial intelligence (AI). Mimicking the human brain is not the major goal of deep learning but deep learning is loosely inspired by the human brain [27]. At present, with the improvement of hardware performance, deep learning is applied as an important method to a variety of practical research and engineering, fasks including security monitoring, business data analysis, and biological information analysis.

2.4.1 Deep learning

ML is divided into supervised learning, unsupervised learning and reinforcement learning. Supervised learning derives the prediction function from the labeled training data. Labeled training data means that each training instance includes an input and the desired output [76]. This method is the most widely type of ML used in computer vision [76]. Unsupervised learning infers conclusions from unlabeled training data. The most typical unsupervised learning method is clustering, which can be used to discover hidden patterns or group data during the exploratory data analysis phase [76]. Reinforcement learning (RL) is a type of ML that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences [76].

The application of supervised learning in the field of computer vision is divided into several categories: image classification, image detection, semantic segmentation and image generation. Among them, image classification distinguishes different types of images according to the semantic information in images, which is an important basic problem in computer vision [40]. A Convolutional Neural Network (CNN) is a feedforward neural network that uses convolution operations in one of its layers instead of matrix multiplication. CNNs are normally used to perform supervised learning. Structurally, it is divided into feature learning and classification. Through the convolution operation, its artificial neurons can respond to a part of the surrounding cells in the coverage area. This feature allows CNNs to handle large images well [1]. For image classification, CNN has very good performance. Its workflow includes four steps. First, input a collection of images and train the CNN. Then, use

convolution to extract features from the image and create feature model. Next, train the model and evaluate the classification results. Finally, save the model for testing. The CNN structure is shown in Figure 2.2:

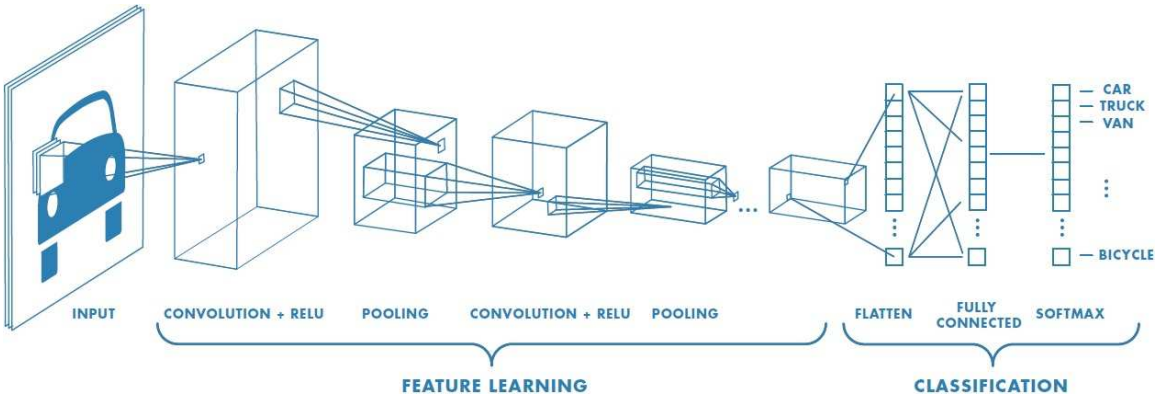

Figure 2.2: The classic CNN structure and principle [68]. This structure is mainly divided into two parts: feature learning part and classification. The feature learning is composed of multiple convolutional and pooling layers, and its main function is feature extraction. The classification part is mainly composed of fully connected layers. The main function is to classify and identify objects. The classifier includes softmax and linear mapping.

2.4.2 Parameters of CNN

In traditional neural networks or deep neural networks (DNNs), different layers of neurons are fully connected, that is, the input of one neuron in this layer will take the output of each neuron from the previous layer [3]. They are called "fully connected neural networks". Full connection has one drawback. The training convergence is

very slow because of a large number of weights and offsets. It takes a long time for fully connected networks when trained on images with millions of pixels to converge and they do not generalize well [3]. CNNs were developed to solve this problem. The operating principles of CNN are introduced as follows:

1. The *Local Receptive Field* (kernel) is used to indicate the size of the sensor's range of perception of the original image at different locations within the network [39, 46]. It is a variable-size window. Compared with the traditional neural network, the convolutional neural network directly transmits image information in layers using local receptive fields. As a result, the size of the image processed by the next layer of the neural network is gradually reduced, thereby increasing the processing speed. The structure of the local receptive field is

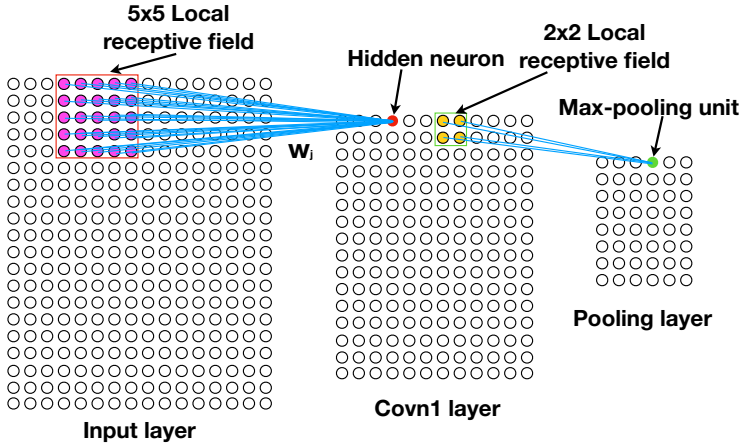

Figure 2.3: Processing of Local Receptive Field [74]

As shown in this figure, the neurons (image pixels) of the input layer are connected to a layer of hidden neurons (called Covn1 layer), where each neuron

is connected to a small region (e.g., 5×5) of the input neurons. The local receptive fields can be set with different sizes, and each neuron in the pooling layer is composed of a Covn1 layer of 2×2 range. Local receptive fields are slid across all neurons in the input layer. For each local receptive field, there is a corresponding neuron in the Conv1 layer [26, 46, 64].

When transmitting image information in the local receptive field, each hidden neuron has a 5×5 weighted kernel connected to its local receptive field in the input layer, and all hidden neurons share the same weights [25]. Weight sharing allows the convolutional neural network to save space and computational complexity when processing images [1, 26, 64].

2. The *Pooling layer* is an optional layer unique to convolutional neural networks. Once the convolution operation is completed, the pooling layer reduces the size of the feature map by using some functions to summarize the sub-regions. These functions are generally for taking the average or maximum value [3]. The structure of this layer is shown in Figure 2.3 and a processing sample is shown in Figure

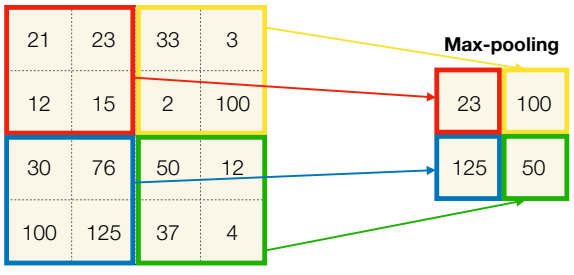

Figure 2.4: Processing of Max-pooling [68]

3. A fully convolutional neural network refers to a CNN that does not have a fully connected layer. Its main advantages are to support the input of different size pictures, support pixel-to-pixel training in the image, and evolve from image level understanding to pixel level understanding. A full convolutional neural network can better learn context information. Feature selection is the goal of this study, so pixel-level processing and classification of images using a fully convolutional neural network can obtain better results. In the full convolutional neural network, the pooling layer is replaced by the discriminal module (see Section 3.5.2 for details). *Batch Normalization* (BN) is an important part of the discriminal module [39, 46]. BN is a method of reducing internal covariate offsets when training traditional deep neural networks [13]. Due to normalization, BN further prevents small changes in parameters from being amplified, and allows for higher learning rates. The structure of BN in CNN is shown in

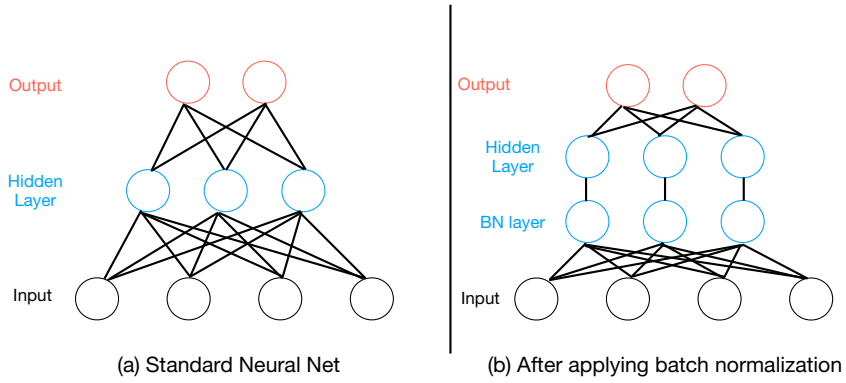

Figure 2.5: The work principle of batch normalization in neural network [36]. (a) is the structure of standard neural net, (b) is the structure after applying BN.

The main idea of BN is to place an extra layer between the layers, where the output of the input layer is normalized [81]. The normalized data is processed by the neurons in the next layer so that the gradient does not change dramatically.

ssing speed

ure 2.6. As

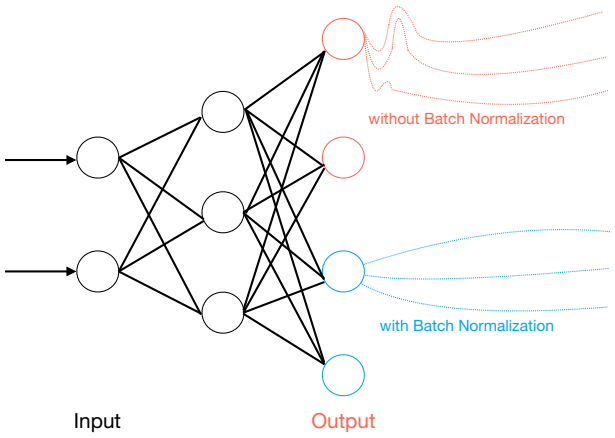

Figure 2.6: The output of batch normalization in neural network [36]. Comparison of the output difference between neural network with BN and without BN, the output of gradient changes severely without BN.

shown in this Figure 2.6, we can see that the gradient change curves of output with BN are smooth (this is expected result), and the gradient changes are obvious as output without BN. BN transforms are typically applied to nonlinear operations in CNN processing (e.g., ReLU).

4. Each neuron in the neural network is weighted and summed, and processed by a function to produce an output, this function is called the activation function. If

the activation function is not used, the output of each layer is a linear function of the input. No matter how many layers are in the neural network, there is a linear relationship between the output and input of either layer [1, 26]. Because the real data processed by the convolutional neural network is nonlinear, it is necessary to introduce nonlinear operations. *Rectified Linear Units* (ReLU) is an activation function for nonlinear operations [64]. Its operation is: if the input value is negative, the output is all 0; if the input value is positive, the output is the same as the input value [26].

Th 7.

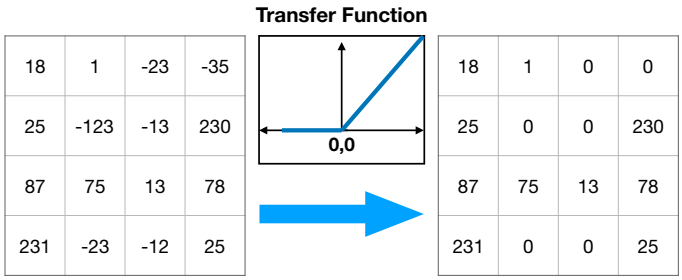

Figure 2.7: Processing of Rectified Linear Units [9]

The purpose of ReLU is to introduce non-linearity processing in CNN since most of the real-world data we want CNN to learn is non-linear. Other non-linear functions such as *tanh* or *sigmoid* can also be used instead of ReLU, but ReLU has been found to perform better in most cases [1, 26, 64].

5. In practical applications, CNN over-process noise and features that are not representative in the training examples, so that the CNN training model cannot accurately extract the features of the sample, and overfit occurs [1]. Overfit

refers to a good performance on the training set but poor performance on the test set, and is indicative of poor generalization performance [1].

Dropout is a way to reduce overfit. Its main idea is that during CNN forward propagation, Dropout randomly removes neurons to prevent weights from converging to the same location [86]. Therefore, the CNN does not need to handle all the weights because of using dropout, which also avoids redundancy. After completing the training, the loss function will open all nodes (include han-

he

in

on

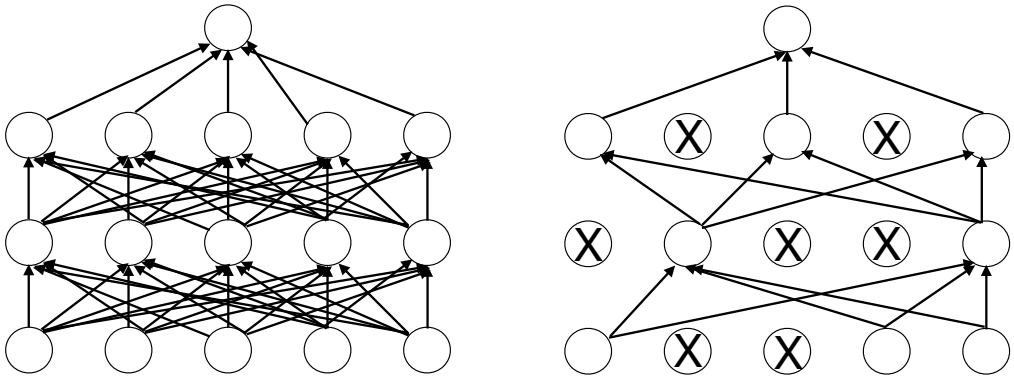

Figure 2.8: The work principle of dropout in neural network [87]. The comparison between using Standard Neural Net and applying dropout, the number of parameters passes between layers is reduced.

of the network or classification of test instances.

6. *Softmax* is the last layer of the convolutional neural network. After CNN feature extraction reaches the final classification layer, the image data becomes 1D sequence data after a series of convolution and merge operations, and then the classifier will classify this 1D sequence data [21, 40, 84]. In general, there are some traditional and useful classifiers including linear classifiers, support vector machines, and Softmax. Softmax is a general-purpose classifier [5].

   Its function is to calculate the probability by the function of Softmax. The output of Softmax is a sequence, and each value in the sequence is the probability of each class [48]. For the sample corresponding to the real label (ie, the category name), the category pointed to by the highest probability value is the classification result of the sample [25, 51, 86].

   The structure of the Softmax classifier is shown in Figure 2.9. In this figure

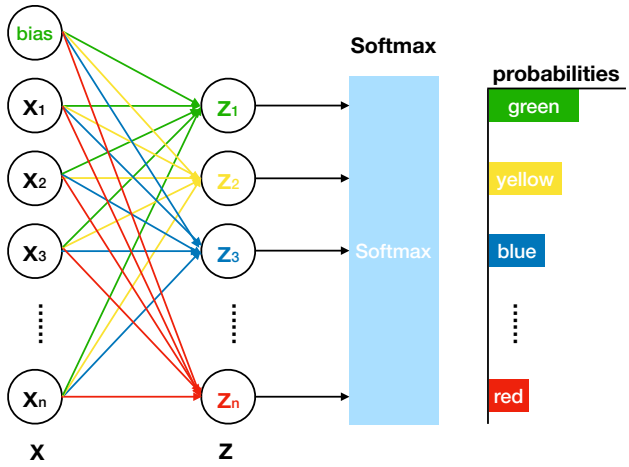

Figure 2.9: Structure of Softmax [68]

above, w_j is full collection parameter (the weight), x_j is the input (the feature),

and z_j is the result after processing of fully convolutional networks, so that we have:

$$z_j = w_j^T \cdot X$$

As the name suggests, the Softmax function is a "soft" version of the max function. The parameter P is probabilities, and the function is

$$p_j = \frac{e^{a_j}}{\sum_{k=1}^{T} e^{a_k}}$$

where a_j represents the j^{th} value in the fully connected parameter w, a_k represents the k values in the vector of the fully connected parameter w, there is a summation symbol (where the sum is k from 1 to N, the range of j is 1 to N, and N is the number of categories). The P value determines the result of the classification, in the sample of probabilities' part of Figure 2.9, the max P value means *green*, the minimum P means *red*.

In practical applications, a typical CNN model includes convolution operations, maximum pooling, and full join operations [64] (as technology updates, many new features and modules are added to CNN), and each step can be repeated many times in order to accurately adjust and train the operation of image compression until the probability of outputting the classification result is maximized [1]. Therefore, CNN can extract more effective features by adjusting various parameters.

2.5 Summary

In this chapter, we introduced some methods and algorithms which will be related to this research includes quality control, data encoding based on space-filling curves,

various feature selection methods and CNN. For data preprocessing of bioinformatics big data, the basic concepts and steps of quality control were introduced. Then, the concept and application of data visualization were described, and several typical visualization algorithms and composition methods for biological information were introduced. Next, for the feature selection method of big data, we simply described the filter algorithm and wrapper algorithm. Finally, the basic structure and working principle of the convolutional neural network to be used were elaborated in detail, and the function of each module of the convolutional neural network was introduced. In the next chapter, we will elaborate on the specific algorithms used in this research.

Chapter 3

Methods

Colorectal cancer (CRC) has received many research efforts on identifying its genetic markers. There exist multiple massive consortia on GWAS for CRC, including the Colorectal Cancer Transdisciplinary (CORRECT) Study, the Colon Cancer Family Registry (CFR), the Molecular Epidemiology of Colorectal Cancer (MECC) Study and the Genetics and Epidemiology of Colorectal Cancer Consortium (GECCO) [71]. We combined two CRC GWAS datasets collected from the province of Newfoundland and Labrador and used new methods to elucidate previously undiscovered susceptibility loci for CRC [71]. There are two datasets processed in this project [71], and the detail is shown in Table 3.1: As there are n samples and each sample consists

Table 3.1: The pretreatment Dataset of CRC [71]

Dataset Name	Individual SNPs	Controls	Cases	Total Samples
Dataset 1	1,236,084	418	278	696
Dataset 2	1,134,514	0	656	656

of m SNPs, then the original SNP data set can be expressed as a $n \times m$ matrix M. The rows of M represent SNPs, and the columns represent samples.

3.1 Work principle of hybrid method

The work flow of proposed methodology is in Figure 3.1. The major work principle

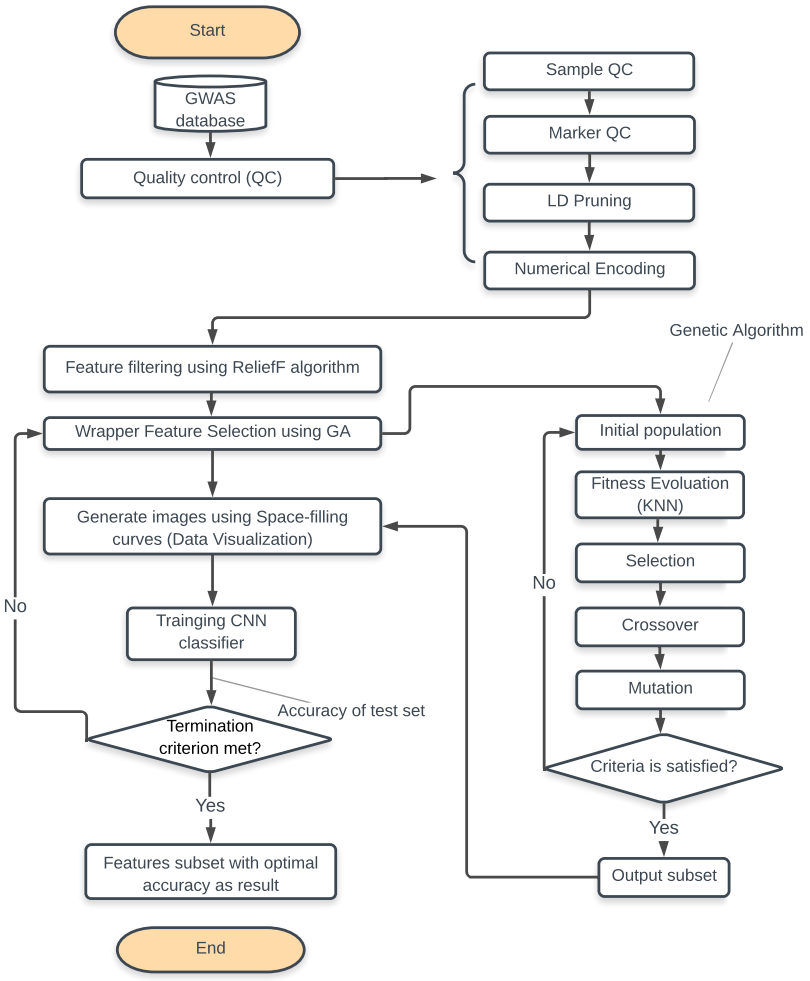

Figure 3.1: The Hybrid Feature Selection Method

of hybrid methods in this research includes five important steps: First, we delete incorrect and sub-standard SNPs and samples through quality control processes and implement numerical encoding for SNPs as the end of quality control. Then, we select the most relevant SNPs (we select the SNPs with higher than the threshold value accord to the requirement in this research) using the filter algorithm ReliefF. Using a hybrid algorithm combining a genetic algorithm for wrapper feature selection, and generating the images based on data visualization (space-filling curve) for sub-dataset that was obtained from GA, we use a CNN model as the classifier of GA to recognize sample categories based on these images. Next, we found a sub-dataset with the highest accuracy of classification for recognizing samples categories, and use this sub-dataset as the final feature selection result. After testing the performance of the hybrid method this research, we determine whether this method is effect. Finally, we provide a final list of genetic markers that can best discriminate between healthy (controls) and CRC cases.

3.2 Data preprocessing

To correctly use SNP data as input for machine learning methods, an extensive set of controls and standards must be put in place. These operations make it possible to detect and rapidly correct many issues, such as technical problems, operator error, inhibitory effects, and other anomalies which lead to inaccurate results [71].

3.2.1 Elements of Quality Control

The software PLINK was used for Quality Control in this research. PLINK is a free, open-source whole genome association analysis toolset, designed to perform a range of basic, enormous-scale analyses in a computationally efficient manner [63], and all thresholds value of quality control in the thesis are referenced from [71]. Quality Control includes some important steps as follows:

1. **Dataset merging** means that two datasets are obtained from health institute, merging these data into one dataset so that more samples can be included in the bioinformatics analysis.

2. **Sample quality control** includes the following steps:

   Remove samples with discordant sex information

   The first step is checking and ensuring that the gender of a subject matches with their number of X chromosomes. If there are any subjects where the X-chromosome data disagrees with the reported gender, the subjects can be removed from the dataset [43].

   Remove sex and mitochondrial chromosomes because these genes information would not be useful for the feature selection [43].

   Remove samples with heterozygosity rate beyond the mean $\pm 3SD$

   Calculate the heterozygosity value for per individual using this formula:

$$PM = \frac{N(NM) - O(HOM)}{N(NM)}$$

and the average value of PM for all samples:

$$AM = \frac{\sum_{i=1}^{N} PM_i}{N}$$

Calculate the heterozygosity rate difference for per SNP:

$$HR_i = PM_i - AM_i$$

where $O(HOM)$ is the observed number of homozygous genotypes, $N(NM)$ is the number of non-missing genotypes and N is the number of all samples.

Remove samples with missing genotype rates higher than 0.01

If the quality of the DNA sample from an individual is low, there will be a higher rate of uncalled genotypes. This is a marker for poor DNA quality overall. To screen out these subjects we use the *–mind* flag which sets the maximum rate of per-individual missingness [62].

3. **Marker quality control** includes in following steps:

   Remove markers with missing genotype rates higher than 0.05.

   We used the *–mind* flag to remove samples with a high rate of poor genotypes. We currently used the *–geno* flag to remove SNPs that have a high genotyping failure rate in this step. This can be due to poor primer design and non-specific DNA binding to a particular SNP probe [43, 62].

   Remove markers with Hardy-Weinberg $HWE > 10^{-4}$.

   There is a predictable relationship between allele frequencies and genotype frequencies sometimes; The genotype distribution is different from what one would

expect based on the allele frequencies; in this case, genotyping error is one potential explanation. Therefore, we typically check for deviation from Hardy-Weinberg equilibrium (HWE) in the controls for a case-control study. The PLINK commands are used to generate p-value for deviation from HWE for each SNP. and the SNPs with $HWE > 10^{-4}$ are removed [43, 62].

Remove markers with minor allele frequency $MAF < 0.05$**.**

The Minor Allele Fraction (MAF) is the relative frequency in a relevant population of the minor (2nd most common) allele. The distinction between common and rare SNPs is not clear and the difference may depend on the sample size. SNPs with $MAF > 0.05$ are retained in this research [43].

Remove markers with significantly different in missing genotype rate between cases and controls ($p < 10^{-5}$), as this means the p-value is less than threshold value [43].

4. **Linkage Disequilibrium (LD) pruning** includes in following steps:

   Remove markers with correlation coefficient $r^2 > 0.6$**.**

   This is an important step in LD pruning, for a pair of SNPs, where p_i, p_j are the marginal allelic frequencies at the i^{th} and j^{th} SNP respectively and p_{ij} is the frequency of the two-marker haplotypes, Pearson Correlation Coefficient r was given by the following formula [17]:

$$r_{ij}^2 = \frac{(p_{ij} - p_i \cdot p_j)^2}{(p_i - p_i^2) \cdot (p_j - p_j^2)}$$

   Remove related samples with $IBD > 0.25$

IBD means Identity by Descent. $IBD = 1$ means duplicates or monozygotic twins, $IBD = 0.5$ means first-degree relatives, and $IBD = 0.25$ means second-degree relatives. Some variation is often occurs around these theoretical values due to genotyping error, LD and population structure; therefore, we left one from the related set and removed others samples with $IBD > 0.25$.

5. **Numerical encoding**

   Each sample is assigned a categorical phenotype label, $label \in \{case, control\}$ where *case* is a diseased sample, and *control* is a normal sample. The phenotype indicates the sample's affection status where 0 means missing, 1 means unaffected and 2 means affected. $SNP_j\,(j = \{1, 2, ..., m\})$ is the value of the jth SNP in sample X_i, which can be one of the four possible value: 0, 1, 2 and NA. Among them, 0 indicates reference homozygous, 1 indicates mutant heterozygous, and 2 indicates mutant homozygous; NA is a typing failure marker.

3.2.2 Filter method: ReliefF

The ReliefF algorithm is generally recognized as a good filter evaluation algorithm, which can effectively eliminate irrelevant features [19]. The ReliefF algorithm, proposed by Kononenko in 1994, solves many types of problems and regression problems, as well as missing data problems [67]. The core idea of the ReliefF algorithm is that the "good" features should measure the distance between similar samples closer [58], and the distance between different category samples further. The weight value is calculated for each SNP by ReliefF algorithm and a threshold δ is used to filter the most relevant SNPs [67]. This the algorithm is described in Algorithm 1.

Algorithm 1 ReliefF algorithm

procedure ReliefF(A, R, H, M) ▷ A is the feature set, R is the sample set, H are near samples of R (near hits), M are nearest samples.

  Input: Training dataset D, iteration number m, number of near samples k

  Initialize threshold value δ

  Initialize weights for all features (A): $W[A] = 0$, $A = 1, 2, ..., p$ ▷ p is number of A

  while $i = 1 : m$ **do** ▷ m is iteration number

    Select a **random sample** R_i from D

    Find k **near hits** H_j same class with R_i

    Find k **near misses** $M_j\,(C)$ in different classes with R_i ▷ for each class $C \neq class\,(R_i)$

    while $A = 1 : p$ **do**

      Update each feature weight: $W[A] = W[A] - diff_H + diff_M$

    end while

  end while

  Select the features with weights more than δ

  Output a data subset S_{new} with n features ▷ features with top n weights

end procedure

From Algorithm 1, the steps of the ReliefF algorithm are as follows: First, the training data set D is input, the iteration number m, the number of samples adjacent to the k, and then each feature weight in the feature set A is initialized as $zero$. Next, the feature weight calculation is performed m times by iterative operation: one sample R_i is randomly taken out from all the samples D. In the sample group of the same classification as the sample R_i, k nearest neighbor samples H_j are taken out. Then, among all other sample groups that are differently classified from the sample R_i, k nearest neighbor samples M_j are also taken out, respectively. The weight of each feature A_p is calculated, and the corresponding weight for each feature is $W[A_p]$. The weights of the p features in feature set A are updated in sequence, the feature difference $diff_H$ of the same class is subtracted, and the difference $diff_M$ of the feature of different classes is added [45, 85]. (If the feature is related to the classification, the values of the feature of the same classification should be similar, and the values of the different classifications should not be similar).

After completing the iterative loop, the weights of each feature are obtained, and each feature weight is compared with the threshold $\delta = 0.5$. If the feature weight is less than δ, it indicates that the feature has a higher degree of independence and is less associated with the disease and should be filtered out during the filter stage [78]. Otherwise, we keep it and place into the output subset S_{new}. Finally, S_{new} is used as the output of the ReliefF algorithm. Some important formulas are expressed as follows:

The difference between hit samples:

$$diff_H = \sum_{j=1}^{k} \frac{|R_i - H_j|}{max(A) - min(A)}$$

The difference between miss samples:

$$diff_M = \sum_{C \neq = class(R_i)} \frac{|P(C)|}{1 - P(class(R_i))} \sum_{j=1}^{k} \frac{\left|R_i - M_{C_j}\right|}{max(A) - min(A)}$$

Feature weight update formula:

$$w = w - \frac{diff_H}{k} + \frac{diff_M}{k}$$

3.3 Wrapper method: Genetic Algorithm

A Genetic algorithm (GA) is a search heuristic that is inspired by natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring in the next generation [8]. The process of natural selection starts with the selection of fittest individuals from a population. They produce offspring which inherit the characteristics of the parents and will be added to the next generation. If parents have better fitness, their offspring will be better and have a better chance at surviving. This process keeps on iterating and at the end, a generation with the fittest individuals will be found [8].

3.3.1 Overview of Genetic Algorithm

Genetic algorithms have shown their adaptability, independent domain knowledge, parallelism, better dealing with enormous-scale complex data, particularly being suited to solving multi-objective optimization problems [8]. Therefore, GA is an ideal

algorithm for wrapper feature selection in a hybrid prediction model for analysis of high dimensional data. A Genetic algorithm does not evaluate the characteristics of a single feature but assesses possible feature subsets. There are various parameters that will be modified during the implementation of using the algorithm to analyze the dataset. A Genetic algorithm can be implemented using the pseudo-code as shown in Algorithm 2.

From Algorithm 2, we can see the execution of the genetic algorithm: First, it inputs the data set S_{new} which was obtained from the previous step, sets the maximum iteration number $Maxiteration$, and initializes $i = 0$ as the iteration initial value. Next, the fitness function is defined by $F(i)$ (we set the classification accuracy as the fitness function) [58]. An initial population $P(i)$ is generated, and the crossover probability p_c and the mutation probability p_m are initialized. Then, the iterative process of the genetic algorithm is performed under the condition that the number of iterations i is less than the maximum iteration number $Maxiteration$ and the best fitness value $bestfitness$ is less than the maximum fitness value $Maxfitness$. As each iteration starts, the iteration number i will be increased by 1. In the following step, the previous iteration population $P(i-1)$ is cross-processed to generate the cross-population $P(i)$ of this iteration [58]. The mutation processing of the population $P(i)$ produces a mutated population of this iteration. The fitness function $F(i)$ is calculated for this mutation population, and the result of this fitness function is the classification accuracy of the selected features [8]. At the end of the iterative loop the output subset Z with the highest accuracy is the final feature selection result. From this algorithm description, we get the genetic algorithm to process and upgrade the data set step by step in each iteration. In this gradual evolution, the final ideal result

is obtained [8].

Algorithm 2 Genetic Algorithm

procedure GENETIC ALGORITHM(P) ▷ P is population
  Input the dataset S ▷ S is data subset from ReliefF
  Setup $MaxIteration$ ▷ $MaxIteration$ is maximum number of iteration
  Initialize iteration $i = 0$ ▷ i is the number of iteration
  Define fitness function $F(i) = ComputeFitness(P(i))$ ▷ The data subset has the best classification accuracy in this generation
  Generate the initial population $P(i)$
  Initialize the probabilities of crossover (p_c) and mutation (p_m)
  while $i < MaxIteration$ **and** $Bestfitness < MaxFitness$ **do**
    $i = i + 1$
    $P(i) \leftarrow Crossover(P(i-1))$
    $P(i) \leftarrow Mutation(P(i))$
    $F(i) \leftarrow ComputeFitness(P(i))$
  end while
  Output final feature set Z ▷ This is final feature selection results
end procedure

3.3.2 Design of Genetic Algorithm

The parameters of our GA are configured as shown in Table 3.2. Here, the termination condition of the algorithm is controlled by the maximum number of generations

and the number of repetitions of the optimal solution. The maximum number of generations is generally valued from 500 to 1000, in order to prevent the occurrence of too many generations or non-convergence. When the fitness value is repeated multiple times, the evolution stops. If this condition is not reached at all, the evolution iteration will be stopped when the maximum number of iterations is reached; we set the number of iterations to be 20. After these preparatory processes are completed, an initial subset of SNPs is generated. For these subsets, the genetic algorithm performs the following important processing:

Table 3.2: Genetic Algorithm configuration

Parameter Name	Function Selected	Value
Population Size	None	20, 30, 40, 50
Generations	None	500, 600, 1000
Representation	Bit String	None
Selection Function	Tournament	None
Mutation Function	Mutation Uniform	0.1
Crossover Function	Crossover Arithmetic	0.8
Elite Count	None	2, 3, 4
Seed	Rand	1, 3, 5, 7
Fitness repeat	None	20

Crossover means that two pairs of chromosomes swap parts of their genes in some way to form two new individuals. Usually, the Crossover processing in GA includes single point intersection, multi-point crossing (include double-point), uniform crossing

and arithmetic crossing. Arithmetic Crossover was used in this research with the P_c referring to the crossover probability. The obtained SNP subsets from the selection are processed by crossover, and generate new SNP subsets, retaining these new subsets to the next step.

Mutation refers to the change of specific gene values in the individual coding string used to form new individuals. The Mutation operation is an auxiliary method to generate new individuals, which determines the local search ability of genetic algorithms and maintains population diversity. Crossover processing and mutation cooperated to complete the global search and local search of the sample space. The Mutation operator changes the value of the chosen gene with a uniform random value selected between the user-specified upper and lower boundary for this gene. P_m refers to Mutation Probability. In this thesis, a new variant subset is generated from each subset of SNPs based on mutation factors, then the selection process is repeated for these subsets, and recalculate fitness. This processing is looped until the SNP subset with the highest classification accuracy rate is found as the final output.

Fitness function: The GA evaluates an individual (solution) by the fitness function value. The classification accuracy is selected as the fitness function value for GA, and the feature subset with best classification accuracy is the final feature selection [45]. K-nearest neighbours (KNN) is a traditional machine learning method, and it is used for classification and regression analysis. The KNN process is as follows: Given a training dataset, for the new input sample, the K samples closest to the input sample are located in the training data set (i.e., the nearest neighbors in the feature space). For input samples, if the amount of one type categories is the most, then this input sample is classified into that category. The model used by the K-nearest

neighbor algorithm is the partition of the feature space. The Euclidean distance is applied in the KNN algorithm, the size of distance will affect the final classification, so the weighted voting method is more appropriate. The fitness function is define as following:

$$C = FitFunc(X, Y, KNN, EuclideanDistance)$$

$$FitVal = \frac{resubloss(C)}{Num_s - Num_f}$$

where X is the samples dataset, Y is the label set for samples, $FitVal$ returns the fitness value, the parameter Num_s is the total number of samples, $Featindex$ means the indexes of selected ones from SNPs sequence, Num_f is the number of elements in $FeatIndex$, and the function $resubloss(C)$ returns classification error by resubstitution. The parameters of KNN are shown in Table 3.2.

Table 3.3: The main parameters of KNN

Parameter Name	Distance Selected	Value
K-value	Euclidean Distance	5, 7, 9, 11

3.4 Data encoding

The SNPs dataset is the regular sequence dataset. There are several various parameters in these sequences. An important parameter is discover in this research is the physical distance of SNPs which is the number of nucleotides between two genes or two positions on a chromosome [4]. Three different methods are used for encoding SNP data in our study, which are based on different filling methods, different image

sizes, and different space-filling curves respectively.

3.4.1 Data encoding based on different filling methods

This process encodes the SNPs sequence per individual from sequence to 2D images [42], which is shown in Figure 3.2. There is a table in top part of Figure 3.2, and

SNP name	rs3131972	rs12562034	rs11240777	rs4970383	rs1110052	rs13303065
Genome value	A G	A G	A G	A C	G T	C T
Genomic Distance	752721	768448	798959	838555	873558	891059
Numerical value	0	2	1	1	1	1

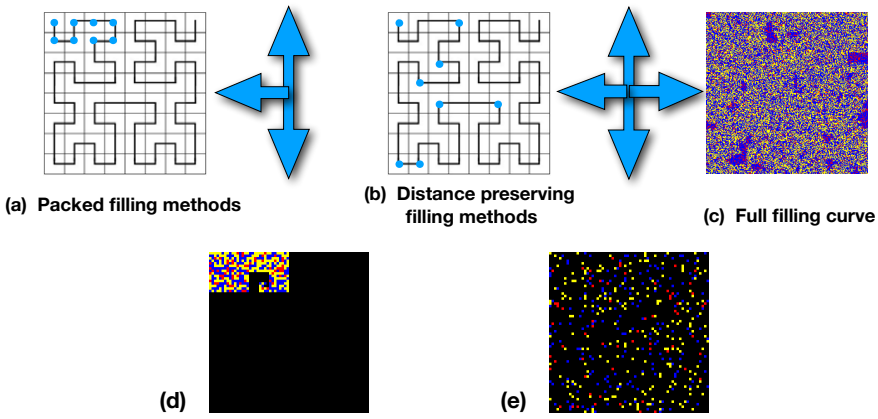

Figure 3.2: The SNPs data encoding based different filling methods

there are some important information of the SNPs sequence after numerical encoding in the table, which includes alleles, Genomic distance of SNP and numerical value (for details, please refer to numerical encoding in section 3.2.1) Furthermore, according to these sequence data, the basic data encoding rules are: First, the sizes of the matrix are determined. In the second step, we compute coordinates for each SNP sequence element along the SFC. Then, we assign the value of each SNP sequence element into

the corresponding location of the matrix, if there is nothing SNP sequence element assign into the matrix; then assign *null* in this location of the matrix. Next, we assign colors to pixels based on the 2D data (matrix), A pixel will be colored as *blue* if the corresponding value is 0, as *yellow* if the value is 1, *red* if the value is 2, and *black* if the value is *null*. Finally, the output images will be generated by the algorithm. Thus, we have:

$$G'(x_i, y_i, n, t) = F(G(x, y), H(p_i), n, t)$$

where G is space-filling curve type and $G(x, y)$ is the location coordinate of features in filling image matrix space based on the space-filling curve, the argument t is which filling method will be selected for SFC, H is the SNP sequence, p_i is the SNP value of number i in this sequence, n is the rank of this space-filling curve and $n \in 1, 2, 3, ..., N$, F is the function to find elements in the SNPs sequence and map them into the corresponding pixel location of an image by the filling method and the SFC type. G' is generated after processing function F on G.

The SNP sequence fragment is the values of the parameter, Numerical value, in Figure 3.2. Figure 3.2 (a) shows the locations of this SNP sequence fragment which is visualized by SFC (Hilbert curve) and the distance preserving filling method. Figure 3.2 (c) is the whole SNP sequence visualized by using the full filling curve method and SFC(Hilbert curve), Figure 3.2 (b) shows the location of the data fragment after mapping the image of Figure 3.2 (c) by using the SFC (Hilbert curve), which is the distance preserving filling methods, and this preserving distance is calculated based on combining features map and physical distance of SNP. Figure 3.2 (d) is (a) after the coloring process, and Figure 3.2 (e) is a combination of (b) and (c) after the

coloring process.

3.4.2 Data encoding with different image sizes

Due to the physical distances of SNP considered, the final images of the same data sequence within different image sizes are different [41]. The SNPs sequence filling with different image sizes is shown in Figure 3.3. The images that are shown in

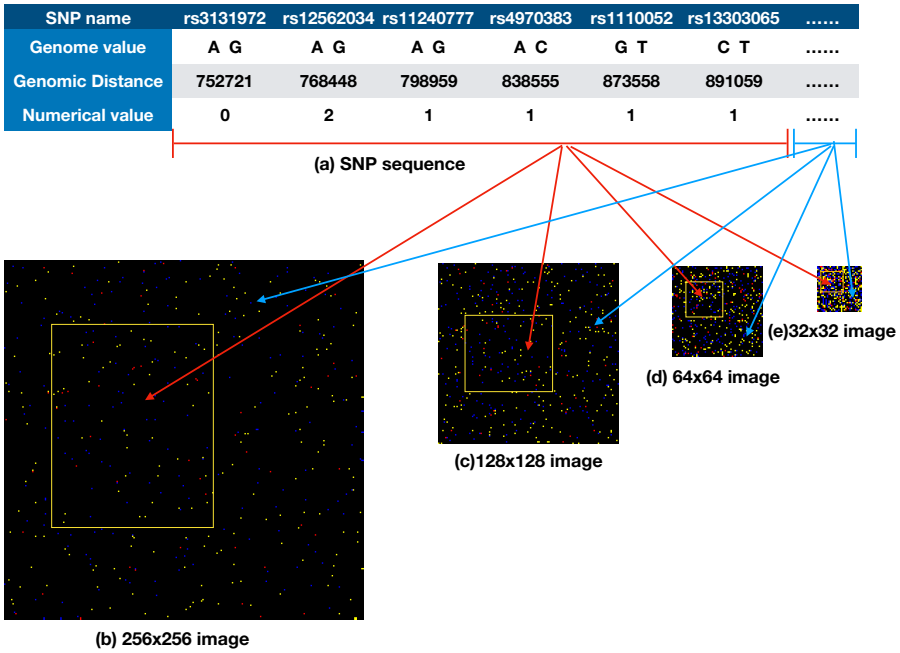

Figure 3.3: SNP data encoding with different size images based on distance preserving filling methods

Figure 3.3 (b), (c), (d) and (e) are colored by same SNP data fragment shown in Figure 3.3(a). All four of these images are obtained after filling with SFC; however, the same data fragment is visualized differently in the different image sizes because

of different spacing calculated from physical distances of SNPs. The feature subset in the yellow window is the red segment of the SNP sequence, and the remaining SNP sequence segments are represented as features outside the yellow window. The images of different sizes are generated from SNPs based on the Hilbert space-filling Curve and two different filling methods, which are shown in Figure 3.4:

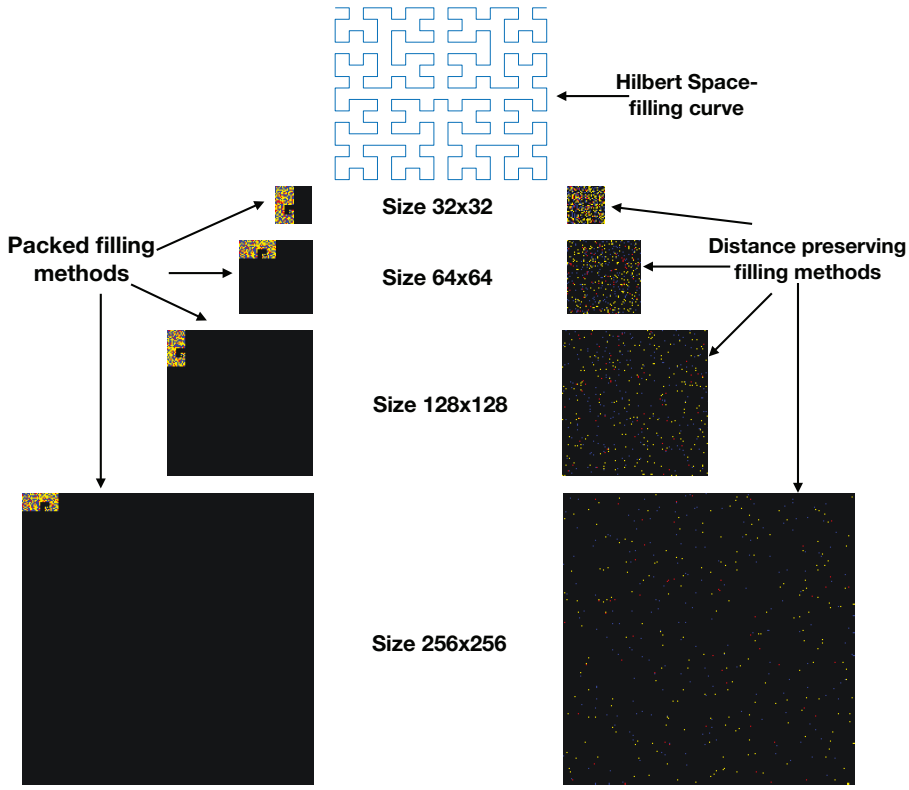

Figure 3.4: The data encoding with different size and filling methods based on Hilbert Curve

3.4.3 Data encoding based on different Space-filling curves

There are five different SFCs used in this research: Cantor, Hilbert, Row, Row-Prime and Spiral curve [57]. The same SNPs sequences are filled based on different SFC and distance preserving or packed filling methods. The results of SFC are shown from Figure 3.5 to Figure 3.8. The Cantor space-filling curve is shown in Figure 3.5: The left figure is the redistributed curve in the matrix of the image, the middle

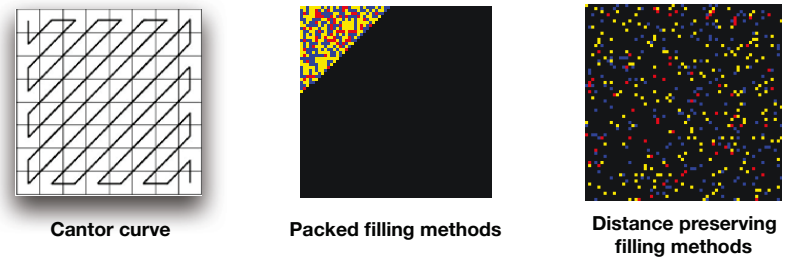

Figure 3.5: Data encoding with different filling methods based on Cantor Curve

figure is the image based on packed filling methods, and the right one is based on distance preserving filling methods. The correspondingly coding results using Row Space Space Filling Curve, Row-Prime Space-filling Curve, and Spiral Space-filling Curve, are shown in Figure 3.6, Figure 3.7, and Figure 3.8, respectively.

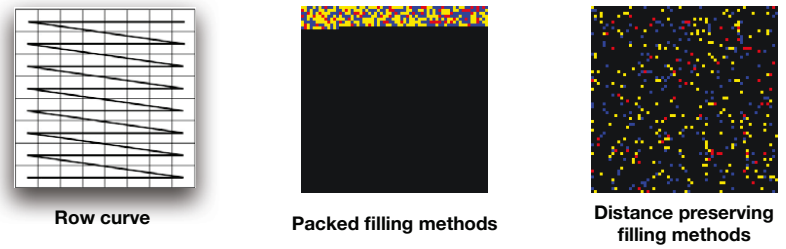

Figure 3.6: Data encoding with different filling methods based on Row Curve

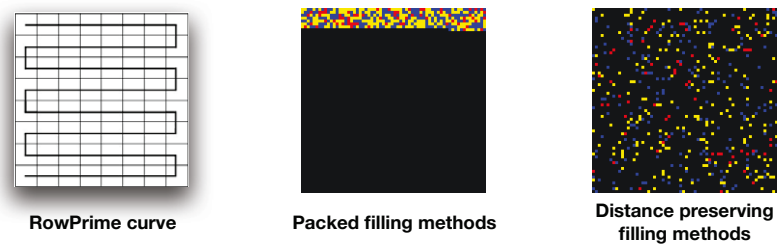

Figure 3.7: Data encoding with different filling methods based on Row-Prime Curve

Figure 3.8: Data encoding with different filling methods based on spiral Curve

3.5 Classification using CNN

3.5.1 CNN as classifier

After obtaining the subsets of SNPs using the feature selection algorithm discussed in Section 3.3, these subsets are then encoded into 2D images based on the spatial fill curve method. These images are divided into four groups: case training sets, control training sets, case test sets and control test sets [48]. To achieve more accurate results, the CNN is designed to train, test and classify subsets of data features generated by genetic algorithm (GA) populations. Given the input images encodings, using a CNN

as classifier allows us to find correlations between a disease and a group of genes.

3.5.2 CNN model based on TensorFlow platform

TensorFlow was used as the deep learning platform in this research. It was developed by Google and released as open source in 2015 [64]. It has been widely used in graphic classification, audio processing, the recommendation system and natural language processing [64]. TensorFlow is a portable deep learning framework that supports Linux, Windows, Mac, and even mobile devices [48]. TensorFlow provides a very rich API for deep learning [64]. It can be said that all the APIs provided in the current deep learning framework include basic vector matrix calculation, various optimization algorithms, the implementation of various convolutional neural networks and the basic unit of the cyclic neural network, and the auxiliary tools for visualization. It also has many advantages, such as a high degree of flexibility, multi-language support and comprehensive documentation [53, 75]. The model structure of CNN for the classifier of hybrid feature selection based on Tensorflow in research is shown in Figure 3.9.

We use a discriminator module, but remove all pooling layers in CNN [7]. Then, the pooling layer is replaced with stride convolution in the discriminator module [7]. Also, the Dropout function is applied to prevent overfitting and memorization [53]. Finally, the full-connection layer is removed. After finished all these changes the deep learning model became a full convolutional network. The advantage of full convolutional network is that they can handle any size input (in this research, there will be various images of different sizes being tested), so it is very suitable for tasks such as detection and segmentation [7]. The reason for introducing this process is to

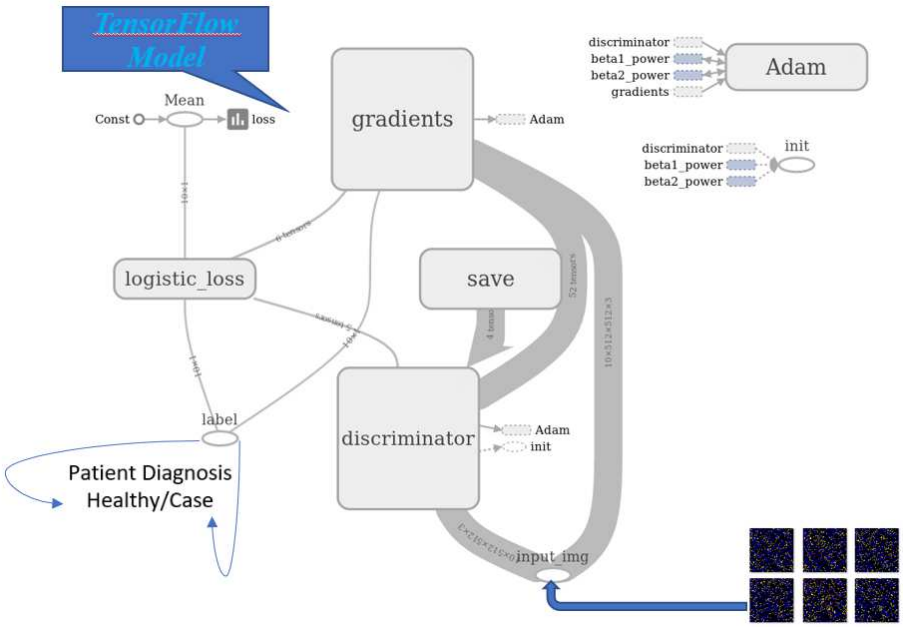

Figure 3.9: The CNN model structure for the classifier of GWAS based Tensorflow. To illustrate the real architecture of CNN for this project, we used the graphs that are generated by TensorBoard to display CNN structure.

verify which SFC can be used, and what size of the SFC is suitable for representing the set of SNPs and the correct description of the disease. Also, some important modules in the CNN structure are as follows:

1. *Input* module is used to input the source images into the Convolutional Neural Network (CNN). An arbiter is required in modules of CNN because the images are processed with various sizes. The convolution and max-pooling layers are modified depending on the image size.

2. Batch normalization is used in the discriminator to solve the problem of gradient disappearance and to solve the problem of unstable data at each layer. Also,

a *Discriminator* module that introduces how real an image is [7] is shown in Figure 3.10; for example, the input is an image (512 pixel x 512 pixel x 3 channel). TThe output is a scalar value, it refers to the authenticity of the disease for this sample (the value would be assumed: 0 is undoubtedly healthy, 1 is undeniably disease case, anything in between is the probability of case) [7].

This evaluation of the authenticity of the generated images in the convolutional layer can train the discriminator to better understand and distinguish the difference between the original images (original 2D-image based on SNPs data encoding) and the generated images [6]. Through this special discriminator process, CNN improves the ability to extract data features, which significantly improves the accuracy of classification and recognition [6]. Appendix A.1 gives the code of discriminator.

3. *Gradients* module is used to compute and update the gradients in every step. The Gradients modules is shown in Figure 3.11:

4. The *Logistic Loss* modules are shown in Figure 3.12: Every algorithm we use in machine learning tasks has an objective function, and these algorithms will implement optimization solutions for the objective function [73]. Especially in the classification problem, the loss function (LF) is used as the objective function of machine learning, also known as the cost function (CF). The loss function is used to evaluate the degree of inconsistency between the predicted values of the model and true values. The smaller the loss function, the better the performance of the model [73]. The purpose of our research is to narrow the difference between reality and forecast (in this research, the forecast value

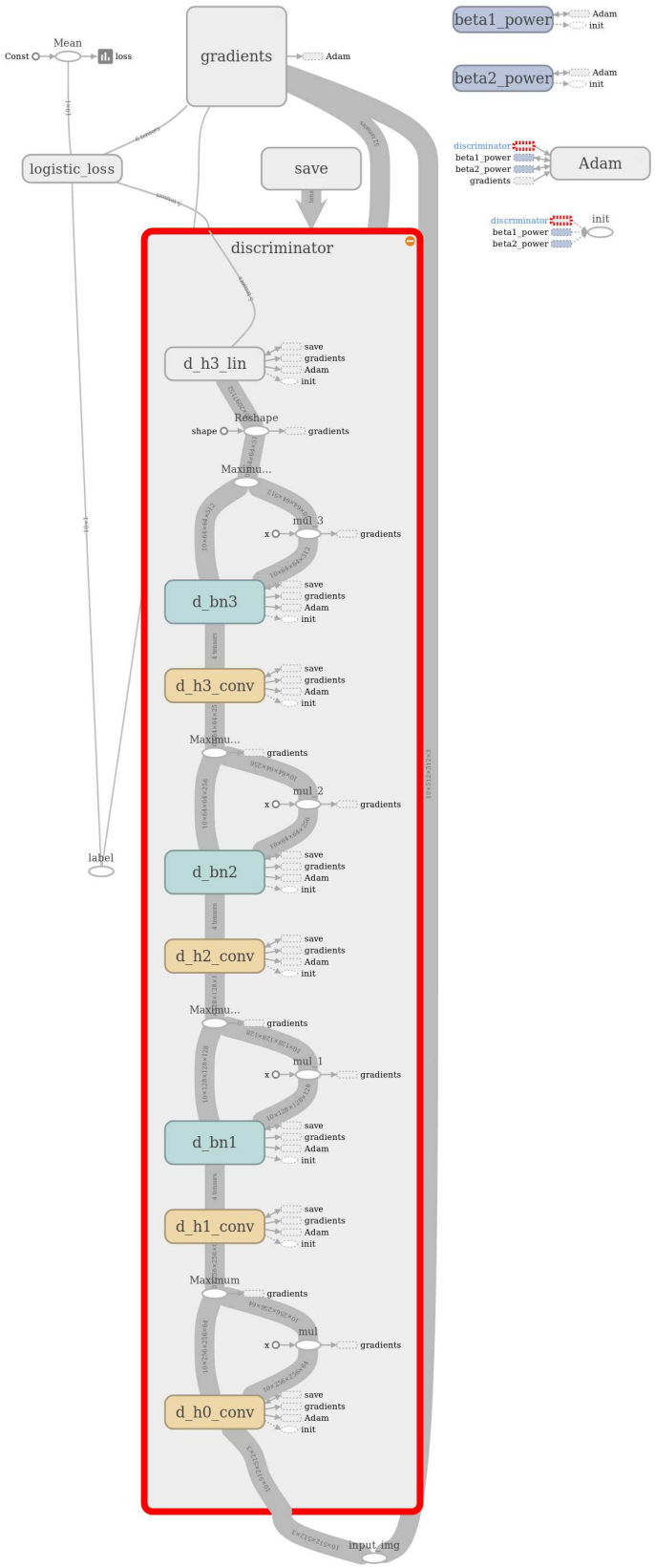

Figure 3.10: The Discriminator Module Structure in Tensorflow

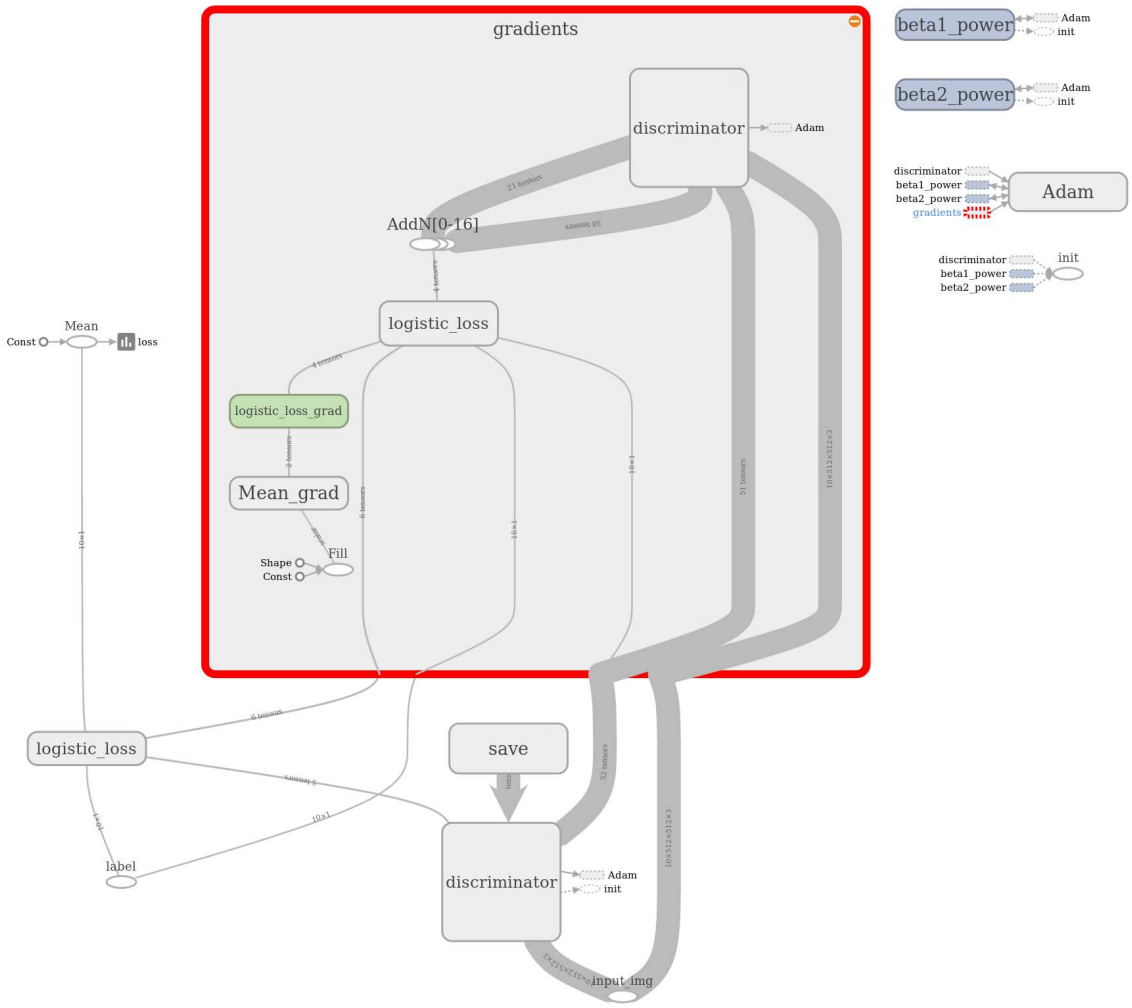

Figure 3.11: The Gradients Module Structure in Tensorflow

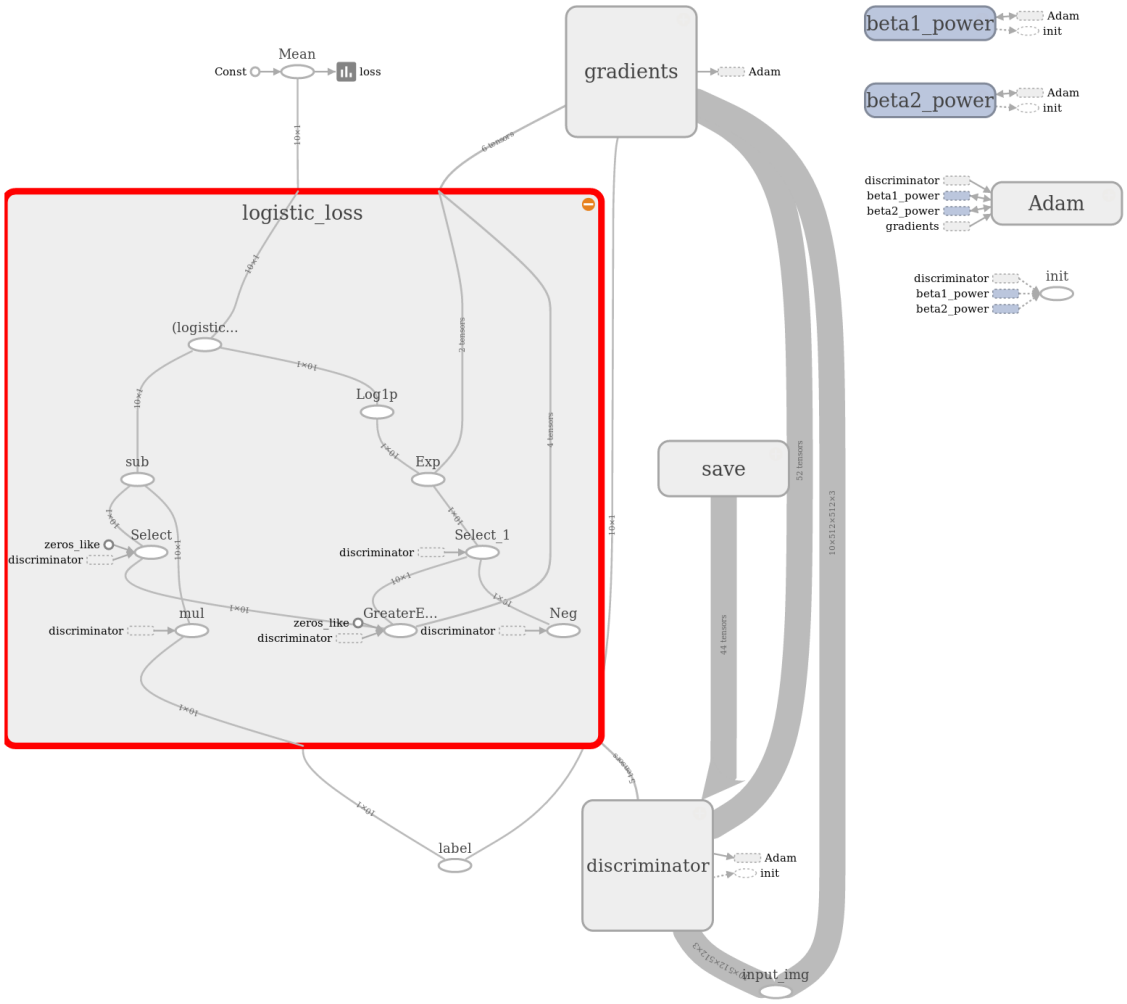

Figure 3.12: The Logistic-Loss Module Structure in Tensorflow

is the health attribute of the sample obtained by CNN classification (healthy or case) and the reality value is the health properties of the sample which are known in advance). The process of continuously reducing the LF value is called optimization [59]. Generally, the loss function is equal to the sum of the loss term (LT) and the regularization term (RT) [59, 73].

We use CNN to deal with classification problems. Additionally we use log loss (Logistic-Loss module in Figure 3.12) as LT [59]. RT is divided into two regularizations: L1 and L2. The L1-regularization refers to the sum of the absolute values of the elements in the weight vector, usually expressed as following formula:

$$\min_{w} \frac{1}{2n_{samples}} \|X_w - y\|_2^2 + \alpha \|w\|_1$$

where w is the feature weight, n is the number of features, $X = \{x_1, x_2, ..., x_n\}$ is the feature sequence, and $\alpha \|w\|_1$ is the L1 regularization term which is expressed as:

$$\|w\|_1 = \sum_{i=1}^{n} |w_i|$$

The L2-regularization refers to the sum of the squares of the elements in the weight vector and then the square root, usually expressed as the following formula:

$$\min_{w} \|X_w - y\|_2^2 + \alpha \|w\|_2^2$$

where $\alpha \|w\|_2$ is the L2 regularization term which is expressed as:

$$\|w\|_2 = \sqrt{\sum_{i=1}^{n} w_i^2}$$

In this research, because the sample space of the data set is not massive enough (less than 2000), using CNN to process such data sets often leads to overfitting when the training data is not enough [59, 73]. As the training process progresses, the complexity of the model increases and the error on the training data decreases gradually, but the error on the verification set gradually increases— this is because the trained network overfits the training set, and the data outside the training set does not work [59, 73].

In order to prevent overfitting, there are many methods that can be used, including early stopping, data augmentation, and regularization including L1, L2 (L2 regularization is also called weight decay), and dropout. We use a combination approach of dropout and L2 norm to work together to solve the overfitting problem. The L1 and L2 regularizations are implemented by modifying the cost function. Dropout is implemented by modifying the neural network itself [86].

L2 regularization used to add a regularization term after the cost function: $\min_{w} \|X_w - y\|_2^2$ represents the original cost function, and the latter term $\alpha {\|w\|_2}^2$ is the L2 regularization term. By deriving the L2 regularization cost function, we can obtain the change in weight w:

$$w \to \left(1 - \frac{\eta\lambda}{n}\right) w - \frac{\eta}{m} \sum_{x} \frac{\partial_{C_x}}{\partial w}$$

As seen from the above formula, the L2 regularization term has the effect of making w "small". In neural networks, regularized networks tend to have smaller weights [59]. In the case of small weights, random changes in data do not have too much impact on the model of, so it is less likely to be affected by local noise

in the data. Without adding regularization, the weights are enormous and the network is more likely to overfit the data and hence is not robust to noise [73].

The L1 function removes a constant, while L2 excludes a fixed ratio of weights. L2 decreases more than L1 If the weight itself is great, and L1 decreases more if the weight is small [59, 73]. L1 will make some of the weights zero which seems to match the idea that only some SNPs will be associated with CRC. However, we need to remove some features with the fixed ratio of weights while implementing feature extraction in each convolutional layer. In this research, the characteristics of the SNP encoding image to be processed by CNN is that a small amount of data with a massive weight, so the L2 regularization function is adopted.

The combination of L2 and Dropout not only solves the over-fitting phenomenon in terms of weight reduction, but also obtains most the correct results in a random default network to solve the over-fitting phenomenon. This combination has a better effect than using only one after we observed the neural network which be used in this thesis. The primary usage of Loss Function (LF) into this research is to adjust the gap between the actual results and CNN predictions by adjusting the loss function values [59], Therefore, the CNN-based deep learning mathematical model can better classify and identify experimental data.

5. *Save* module refers to save the template data and information.

6. *Label* module output the final results from the sequence, which are processed by Discriminator and mapped to the label final results include two types of classification: Healthy and Case. The Healthy label refers to negative, and the

Case label refers to positive.

7. The major study work is about binary classification in this thesis, so assessing the classifier of binary classification is a necessary step [38]. The classification accuracy is not the only standard to evaluate the deep learning model. The performance indicators for evaluating a binary classifier include accuracy, precision, recall, ROC, and AUC [38].

   In binary classification task, P refers to the number of positive samples (case samples), and N refers to the number of negative samples (health or control samples). Therefore, after predicting with the classifier, the instance test results can be divided into four groups of data that are used to calculate various performance metrics [69]. These four Instance test results include True positive (TP) samples, TP is the number of true positive samples, TN is the number of true negative samples, FP is the number of false-positive samples, and FN is the number of false-negative samples [69]. These four results are usually represented in a 2D contingency table or confusion matrix that is shown in Table 3.4.

Table 3.4: Confusion Matrix.

Real samples	True	False
Positive samples	True Positive	False Positive
Negative samples	True Negative	False Negative

Based on these four results, we can get the Accuracy (ACC), which refers to

how close the measured value is to its 'true' value [38]. From the perspective of measurement error, ACC reflects the systematic error of the measured value. But this indicator is not appropriate when the category ratio is not balanced.

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

Precision refers to the proportion of true positives in all positive cases.

$$P = \frac{TP}{TP + FP} \tag{3.2}$$

Recall refers to the prediction which is proportional to the positive case in all true positive cases.

$$R = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{3.3}$$

Our model can output a probability prediction value. A threshold can then be set to separate the examples. Above the threshold, the test sample is considered as positive, below the threshold, the test sample is considered as negative [34]. The process of classification is to set the threshold and use the threshold to cut off the predicted value. When the threshold changes, the prediction result and the confusion matrix will change, which will eventually lead to change of the values of some evaluation indicators. In signal detection theory, a receiver operating characteristic (ROC), or simply ROC curve, is a graphical plot which illustrates the performance of a binary classifier system as its discrimination threshold is varied [34].

Now, we get two indicators from the confusion matrix. One is True Positive Rate (TPR), which represents the proportion of positive cases predicted in all

positive cases [69]:

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{3.4}$$

The other is False Positive Rate (FPR), which indicates the proportion of positive cases predicted in all negative cases [69]:

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN} \tag{3.5}$$

Taking FPR as the abscissa and TPR as the ordinate, the ROC curve is the connection of all the coordinate points (FPR, TPR) obtained after changing various thresholds; this curve is shown in Figure 3.13. As shown in this figure,

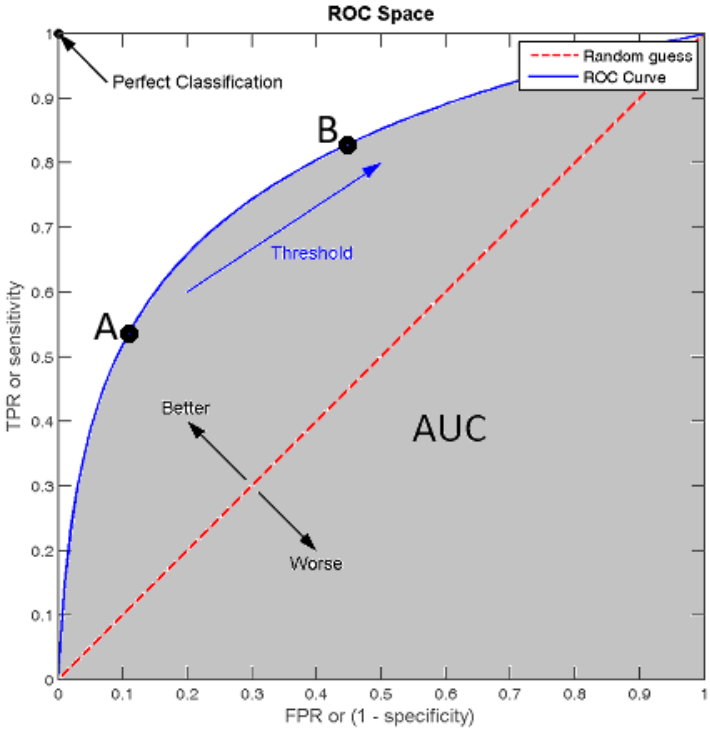

Figure 3.13: Using a ROC curve to understand the diagnostic value of a test, The shaded area value below the curve is the AUC value [88].

the red line is the ROC under random guessing. The more the curve is in the upper left corner, the better the classifier. In the real world, the threshold is discretized because the data is corresponding one by one, and the curve presented is jagged. If there is more data and the threshold is finer, the curve will be smoother [34].

AUC (Area Under Curve) is the area under the ROC curve and it is a probability value [38]. When randomly choosing a positive sample and a negative sample, the AUC value is the probability that the current classification algorithm ranks the positive sample in front of the negative sample based on the calculated score value. Generally, the greater the AUC value, the more likely the current classification algorithm is to rank the positive samples in front of the negative samples, which is a better classification. Generally, we always use AUC to judge the quality of the classifier (predictive model) [38]. The sample figure is shown in Figure 3.14.

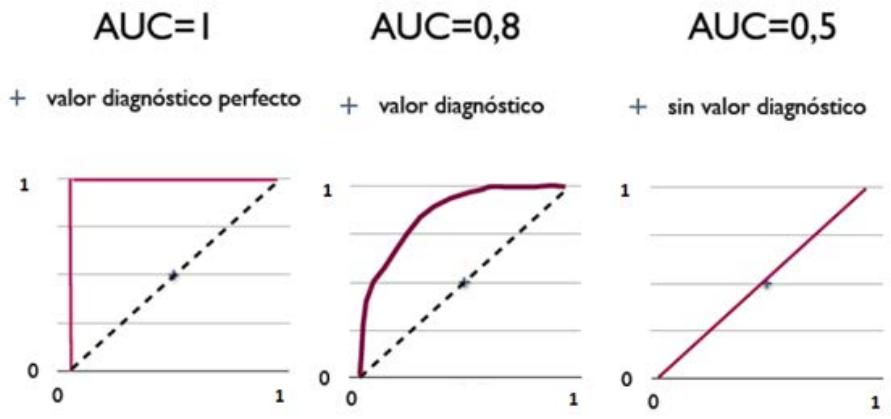

Figure 3.14: The samples for AUC with different value [88].

As shown in Figure 3.14, $AUC = 1$ is an ideal classifier. When using this predictive model, there is at least one threshold to get a perfect prediction. In most cases of prediction, there is no ideal classifier. Normally, the AUC value range is in the interval of $0.5 < AUC < 1$. If the classifier (model) properly sets the threshold, it can have predictive value [38]. $AUC < 0.5$, which is worse than random guessing; however as long as it is always anti-predictive, it is better than random guessing. Simply, the greater the AUC value, the higher the correct rate.

The ROC and AUC are used because the ROC curve has a very good property: the ROC curve can remain unchanged when the distribution of positive and negative samples in the test set changes [16]. In a real data set, there is often class imbalance, that is, the negative sample is much more than the positive sample (Because in reality, the number of colon cancer patients is much smaller than the number of normal people), and the distribution of positive and negative samples in the test data may also change with time [16]. Figure 3.15 is a comparison of the ROC curve and the Precision-Recall curve. In Figure 3.15, it can be clearly seen that the ROC curve remains basically the same, while the Precision-Recall curve changes a lot. Therefore, using ROC and AUC to evaluate performance of CNN is a regular and effective method. We will set up the probabilities P in Softmax classifier of CNN as the threshold value, draw the ROC and calculate AUC based on this threshold value.

3.6 Summary

In this chapter, we elaborated on the specific algorithms used in this study: how to implement each steps for hybrid feature selection methods, which development tools and platforms to use, and which functional modules to use when implementing. First, we executed data preprocessing - quality control of biological information; this process allows the raw data to be processed to get the data we will use. Then, we introduced how to realize the hybrid method of combining all the algorithms to perform our research, including the feature selection method using the ReliefF algorithm combined with a genetic algorithm. In the classification of feature selection, we focused on the method for converting 1D sequence data into 2D images, using convolutional neural networks as classifiers, and implemented convolutional neural networks based on the TensorFlow platform. Finally, we describe how to evaluate the performance of CNN by using ROC and AUC. After completing the implementation of the entire algorithm, we will discuss and analyze the experimental results in the next section.

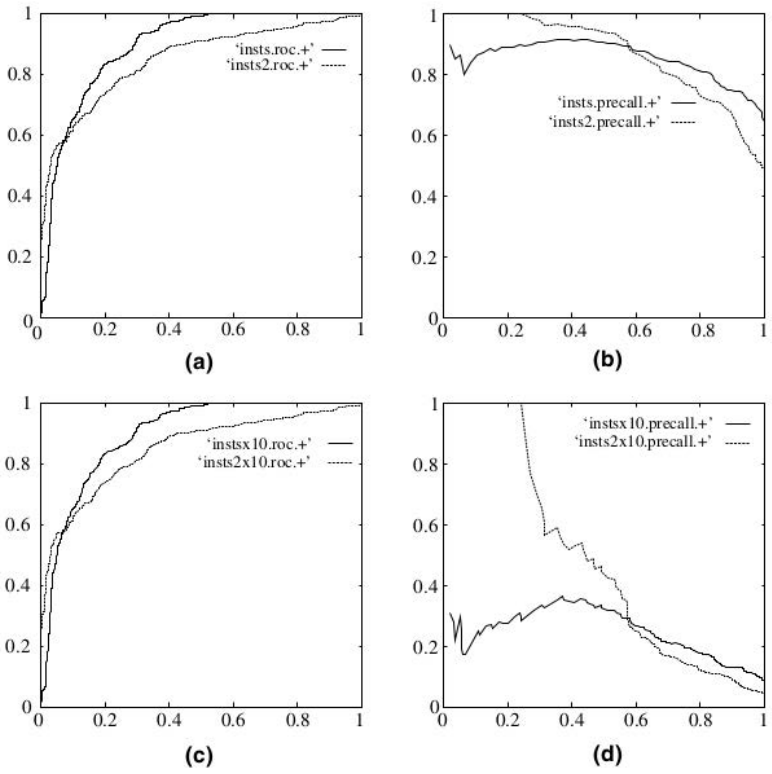

Figure 3.15: The comparison between ROC curve and Precision-Recall curve. The (a) and (c) are ROC curves, and (b) and (d) are Precision-Recall curves. (a) and (b) show the results of classification in the original test set (balance of positive and negative sample distribution), (c) and (d) are to increase the number of negative samples in the test set by 10 times classification [16].

Chapter 4

Results

4.1 Post-processed data

Data preprocessing includes two parts: Quality Control (QC) and filtering. QC includes dataset merging, sample quality control, marker quality control, LD pruning and numerical encoding. We merge two datasets in order to perform execute QC , the processing result is shown in Table 4.1.

4.1.1 Quality Control results

The process of Quality control (QC) include sevral steps: Sample quality control, Marker quality control, LD pruning and Numerical encoding. Each step also includes sub-steps. The results for QC are shown in Table 4.2. The marker quality control results are shown in Table 4.3, and LD pruning results are shown in Table 4.4.

SNP genotypes were coded using three numerical values: 1, 2 and 0. Here, 0 stands for the homogeneous reference genotype, 1 represents the heterogeneous variant, and 2

Table 4.1: Merge two datasets

The first dataset:
1134514 variants loaded from .bim file
656 people (393 males, 263 females) loaded from .fam
Among remaining phenotypes, 656 are cases and 0 are controls
The second dataset:
1236084 variants loaded from .bim file
696 people (418 males, 278 females) loaded from .fam
Among remaining phenotypes, 200 are cases and 496 are controls
The dataset was processed by merging:
486335 variants loaded from .bim file
1352 people (811 males, 541 females) loaded from .fam
Among remaining phenotypes, 856 are cases and 496 are controls

Table 4.2: Sample quality control results for each step

Sub-QC step	Data loaded	Data after QC process
Remove samples with discordant sex information	486335 variants loaded from .bim file 1352 people (811 males, 541 females) loaded from .fam	486335 variants and 1238 people pass filters and QC Among remaining phenotypes, 747 are cases and 491 are controls
Remove samples with heterozygosity rate beyond mean $\pm 3SD$	486335 variants loaded from .bim file 1238 people (709 males, 529 females) loaded from .fam	472727 variants and 1238 people pass filters and QC Among remaining phenotypes, 747 are cases and 491 are controls
Remove the samples with HR beyond mean $\pm 3SD$	472727 variants loaded from .bim file 1238 people (709 males, 529 females) loaded from .fam	472727 variants and 1185 people pass filters and QC Among remaining phenotypes, 702 are cases and 483 are controls
Remove samples with missing genotypes higher than 0.01	472727 variants loaded from .bim file 1185 people (692 males, 493 females) loaded from .fam	472727 variants and 1117 people pass filters and QC Among remaining phenotypes, 639 are cases and 478 are controls

Table 4.3: Marker quality control results for each step

Sub-QC step	Data loaded	Data after QC process
Remove markers with missing genotypes higher than 0.05	472727 variants loaded from .bim file 1117 people (672 males, 445 females) loaded from .fam	472650 variants and 1117 people pass filters and QC Among remaining phenotypes, 639 are cases and 478 are controls
Remove markers with Hardy-Weinberg $HWE > 10^{-4}$	472650 variants loaded from .bim file 1117 people (672 males, 445 females) loaded from .fam	471715 variants and 1117 people pass filters and QC Among remaining phenotypes, 639 are cases and 478 are controls
Remove markers with minor allele frequency $MAF < 0.05$	471715 variants loaded from .bim file 1117 people (672 males, 445 females) loaded from .fam	366094 variants and 1117 people pass filters and QC Among remaining phenotypes, 639 are cases and 478 are controls

Table 4.4: LD pruning results for each step

Sub-QC step	Data loaded	Data after QC process
Remove markers with significant differences in missing genotype rate between cases and controls	366094 variants loaded from .bim file 1117 people (672 males, 445 females) loaded from .fam	366090 variants and 1117 people pass filters and QC Among remaining phenotypes, 639 are cases and 478 are controls
Remove markers with correlation coefficient $r^2 > 0.6$	366090 variants loaded from .bim file 1117 people (672 males, 445 females) loaded from .fam	185180 variants and 1117 people pass filters and QC Among remaining phenotypes, 639 are cases and 478 are controls
Remove related samples with $IBD > 0.25$	185180 variants loaded from .bim file 1117 people (672 males, 445 females) loaded from .fam	185180 variants and 1098 people pass filters and QC Among remaining phenotypes, 626 are cases and 472 are controls

represents the homogeneous variant. There are a total of 1098 samples in the dataset after Quality Control including 472 control samples and 626 case samples, and there are 185180 SNPs.

4.1.2 Filtered data using the ReliefF algorithm

The feature dimension of the data was further reduced using a filter method (ReliefF algorithm) which assigns a weight to each SNP based of its contribution to classifying the disease outcome. Since the Hilbert space-filling curve was used in the process of data visualization, the number of SNPs must be equal to n^2 after using filter method [29], which means that the images were squares. Therefore, the 1024 top ranked SNPs were selected. For classification using CNN, the dataset was divided into a training set and a test set. The proportion of training set to test set is 3:1, including the training set (control number is 354, case number is 470) and the test set (control number is 118, case number is 156).

4.2 Feature selection and classification results

In this section, we analyze the experimental results. In order to show the experimental results under various conditions more prominently, we display the average accuracy and error bars separately.

4.2.1 Comparison of different filling curves

There were five different space-filling curves (SFC) used in this research: Cantor, Hilbert, Row, Row-Prime and Spiral. The results of average testing classification

accuracy are shown in Figure 4.1. The blue histograms are the average classification

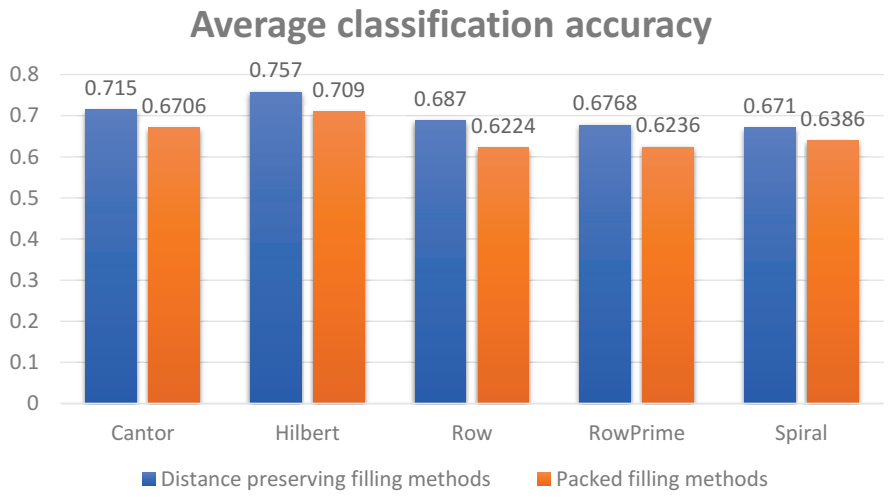

Figure 4.1: The results of average testing accuracy using images generated by different space-filling curves

accuracy using filling curves with distance preserving filling methods if SNPs are distant in the genome. The orange histograms are the results using curves with packed filling methods, that is, selected SNPs are placed one next to another without preserving their distance information from the genome. The Hilbert curve possessed the best accuracy with distance preserving or packed filling methods. For the same space-filling curve, the comparison of results in the histogram reveals that all of the accuracy results based on distance preserving filling methods are better than those based on packed filling methods for the same space-filling curve. The Hilbert curve derived by using the distance preserving filling method possessed the best average accuracy being a value of $75.7\% \pm 0.123$.

4.2.2 Comparison of different image sizes

In this section, the average accuracy results based on different image sizes are analyzed and discussed. Considering that the same sequence was visualized to images based on the same SFC and the same filling method, it is possible to generate different classification results using different image sizes. There were images with five sizes designed through data visualization using all of space-filling curves for the same SNP sequence: 32×32, 64×64, 128×128, 256×256 and 512×512. The average testing accuracy results based on five image sizes are shown in Figure 4.2. The blue histogram

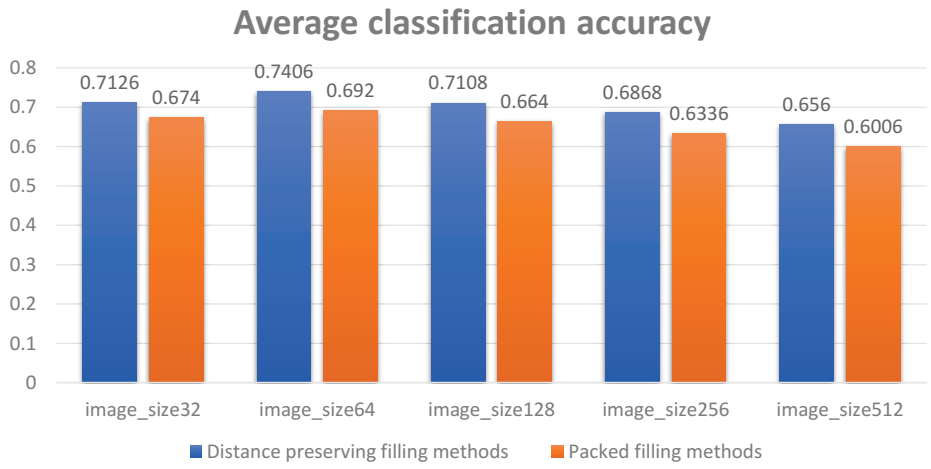

Figure 4.2: The average testing accuracy results using different image sizes based on various space-filling curves

is the average accuracy based on the space-filling curve with distance preserving filling methods, and the orange histogram is the average accuracy based on the space-filling curve with packed filling methods. From a comparison of all results, based on the same space-filling curve, the average accuracy of images with distance preserving

filling methods is better than that of images with packed filling methods. Not only 64×64 images based on distance preserving filling methods possessed better accuracy than others.

4.2.3 Final selected SNP features

The error bars of results are shown in Figure 4.3. The best accuracy appeared in the Hilbert curve with distance preserving filling methods in size 64×64, which is 90.5%. The comparison of results using Hilbert curve are in Table 4.5.

Table 4.5: The comparison of results based on the Hilbert Curve

Amount	SFC	Filling method	Images size	Average accuracy
361	Hilbert Curve	distance preserving	32×32	0.786 ± 0.113
481	Hilbert Curve	distance preserving	64×64	0.842 ± 0.093
512	Hilbert Curve	distance preserving	128×128	0.763 ± 0.124
589	Hilbert Curve	distance preserving	256×256	0.761 ± 0.121
421	Hilbert Curve	distance preserving	512×512	0.752 ± 0.131
384	Hilbert Curve	packed filling	32×32	0.761 ± 0.108
467	Hilbert Curve	packed filling	64×64	0.782 ± 0.113
552	Hilbert Curve	packed filling	128×128	0.738 ± 0.135
581	Hilbert Curve	packed filling	256×256	0.696 ± 0.131
459	Hilbert Curve	packed filling	512×512	0.687 ± 0.157

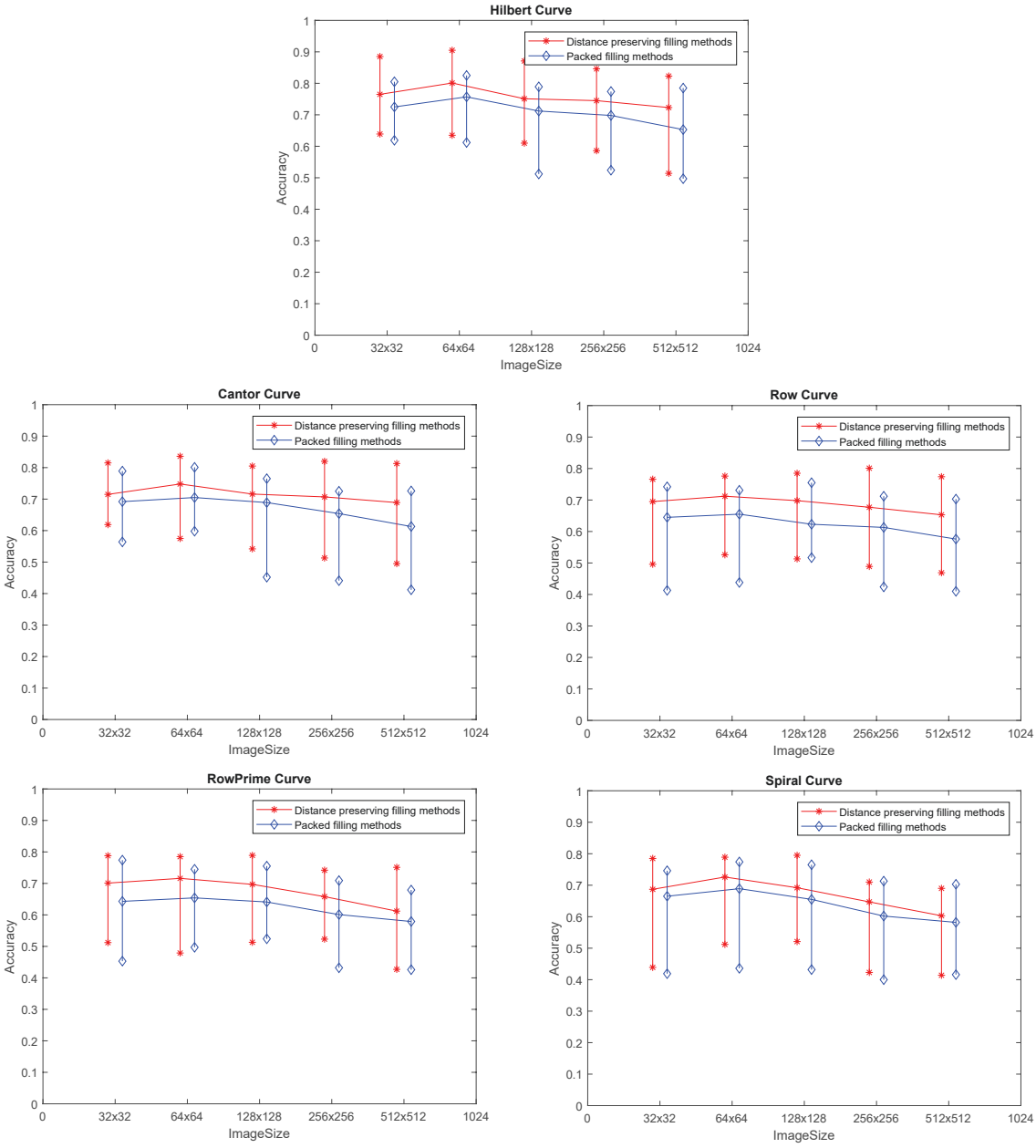

Figure 4.3: The error bars of results based on various SFC with different image sizes

4.2.4 Comparison with other feature selection and classification methods

To validate our proposed hybrid method in this research, several feature selection methods and classification are compared. The results are shown in Table 4.6. The

Table 4.6: The comparison results

Feature selection	Classifier	selected SNP	Data type	Average Accuracy
ReliefF+GA	KNN Fitness	469	Sequence	0.805 ± 0.067
ReliefF+GA	SVM	117	Sequence	0.876 ± 0.024
ReliefF+GA	MLP	405	Sequence	0.861 ± 0.032
ReliefF+SFFS	Naive Bayes	197	Sequence	0.848 ± 0.043
ReliefF+SBFS	Naive Bayes	285	Sequence	0.856 ± 0.047
ReliefF+GA	CNN	481	Image	0.905 ± 0.048

first three traditional hybrid methods are compared in accuracy with each other and with the hybrid method proposed in the current research. These three traditional methods use GA as their wrapper method but use three different classifiers: the K-nearest Neighbors fitness function (KNN), the support vector machines (SVM) classifier, and the multilayer perceptron (MLP) which is a type of feed-forward artificial neural network. The three classifiers all use the DNA sequence as processing data. The second method used the Sequential Forward Floating Search (SFFS). The main idea is using SFFS as a wrapper Feature Selection method with Naive Bayes (NB) as the classifier, and sequence as the data type. The third hybrid method used

the Sequential Backward Floating Search (SBFS). The main idea is using SBFS as a wrapper Feature Selection method with Naive Bayes as the classifier, and sequence as the data type. The classification principle of Naive Bayes classifier is to calculate the posterior probability by the Bayesian formula through prior probability, and select the classification result corresponding to the maximum posterior probability [60]. Given a sample data set D, we have:

$$P\left(c|x\right) = \frac{P\left(c\right)P\left(x|c\right)}{P\left(x\right)} \tag{4.1}$$

where $P(c|x)$ is the probability that the data belongs to the c category for a given feature attribute x, $P(c)$ is the proportion of each category in the sample space, also called it the prior probability, $P(x|c)$ is the probability that the feature attribute combination under a given category of conditions, also called likelihood, and $P(x)$ is the evidence factor for normalization. A very important assumption in the NB algorithm is that each attribute of the data is conditionally independent, because the attributes are independent of each other [60], then we can get a new formula:

$$P\left(c|x\right) = \frac{P\left(c\right)P\left(x|c\right)}{P\left(x\right)} = \frac{P\left(c\right)}{P\left(x\right)}\prod_{i=1}^{d}P\left(x_i|c\right) \tag{4.2}$$

Based on the above, the training process of the Naive Bayes classifier is based on the training set D to estimate the class prior probability $P(c)$, and the conditional probability $P(x_i|c)$ is estimated for each feature attribute. We calculate the probability $P(c|x)$, and get the corresponding classification result [60]. The final hybrid method proposed in this research uses the Convolution Neural Network as the classifier and image as the data type.

A comparison in the accuracy of results reveals that ReliefF+GA using SVM as the classifier and the method proposed in this research had better accuracy. ReliefF+GA

using SVM as the classifier and ReliefF+SFFS could select a lower number of features. However, the SVM only processes the sequence without any other parameters between features. The individual data value and simple linear relation (for example, the sequence data) cannot better represent the complex relationship between biological information data [4]. Therefore, due to the complex relationship between the data in biological information and getting SNPs which are more relevant with colon cancer, we should consider as many factors as possible when dealing with such data, such as associations, physical distances, genomic distance [4]. After comparison all of methods above, the hybrid method in this thesis not only proposes a new SNP analysis method but also has a higher accuracy than others.

4.3 Training performance evaluation

In diagnostic tests or feature selection tests, the diagnostic or selection value of the method is generally evaluated by calculating the corresponding sensitivity, specificity, false positive value, false negative value, and accuracy.

4.3.1 Performance metrics

Sensitivity and specificity are statistical measures of the performance on a binary classification test [18], also known in statistics as classification function. There are two performance metrics, specificity and sensitivity, are used to describe the performance of the classifier. Sensitivity (also called true positive rate) refers to the proportion of samples that are actually positive and are considered to be positive. Specificity (also called true-negative rate) is the ratio that is regarded to be negative in a sample that

is actually negative [69].

Based on the confusion matrix, we got the final result with optimization accuracy after optimizing and comparing all results under various conditions. The final results' matrix is shown in Table 4.7

Table 4.7: Confusion Matrix.

True Positive (138 samples)	False Positive (18 samples)
True Negative (109 samples)	False Negative (9 samples)

The classification results of this research were obtained under the following conditions: CNN as classifier, images generated by the Hilbert space-filling curve, distance preserving filling methods, and image size of 64×64. We assumed case as positive and control as negative. Depending on this confusion matrix (the results with optimization accuracy), we can obtain some important binary classification test results. The Sensitivity or True Positive rate (TPR), the function is shown as below:

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} = 88.4\% \pm 0.026 \tag{4.3}$$

The Specificity or True Negative rate (TNR) refers to the percentage of patients who are actually diagnosed as disease-free without being sick. The function is shown as below:

$$TNR = \frac{TN}{N} = \frac{FP}{TN + FP} = 92.6\% \pm 0.032 \tag{4.4}$$

The Miss rate or False Negative rate (FNR) function is shown as below:

$$FNR = \frac{FN}{P} = \frac{FN}{TP + FN} = 1 - TPR = 11.6\% \pm 0.021 \tag{4.5}$$

The Fall-out or False Positive rate (FPR) function is shown as below:

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN} = 1 - TNR = 7.4\% \pm 0.013 \tag{4.6}$$

The overall function is shown in below:

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN} = 90.5\% \pm 0.048 \tag{4.7}$$

4.3.2 Assessment performance of CNN by ROC and AUC

To assess the performance of CNN completely, we used ROC and AUC. We select performance evaluation of CNN based on the Hilbert curve with best accuracy results (Section 4.3.1). The evaluation results and curve are shown in Figure 4.4.

Figure 4.4 shows the ROC and AUC results based on the Hilbert curve with various conditions because the best average accuracy is obtained using the Hilbert curve. After comparison of all ROC and AUC based on all curves and sizes with different filling methods, we found the largest $AUC = 0.8960 \pm 0.0244$ is generated from the Hilbert curve with a 64×64 image with distance preserving methods. The ROC and AUC is shown in Figure 4.4(a). The red area represents $AUC < 0.5$ and the green area represents $0.5 < AUC < 1$. All ROCs are above the diagonal, and all AUC values are between 0.5 and 1 in Figure 4.4. Therefore, we select Figure 4.4(a) that has the greatest AUC value finally, indicating that CNN is the most effective as a classifier under this condition [38].

4.3.3 Assessment of statistical significance

We use permutation test to assess the significance level of our classification result. The permutation test is based on a large number of calculations (computationally

intensive) by using full (or random) sample data, and is a statistical inference method due to its overall distribution of freedom [23]. The Permutation test has been widely used, especially for the overall distribution of unknown small sample data, and the difficulty using traditional methods to analyze data to test hypotheses [23]. Regarding the specific use, by randomly shuffling the sample labels, the statistical test quantity is recalculated, the empirical distribution is constructed, and then the p-value is calculated and inferred [23].

Disrupt the grouping of samples which is already labeled, randomly group dataset. This new dataset was categorized using the classifier of this research. The above operation was repeated 1000 times. From these 1000 test results, we observed that are 26 classification accuracies exceeding the highest value 0.905 in this research. Therefore, the obtained permutation test value was: p=0.026.

4.3.4 Final feature selection results description

The bioinformatics details of the SNPs obtained by the hybrid feature selection method are shown in the Figure 4.5. This figure shows the SNPs gene information after feature selection. Every histogram denotes how many SNPs are selected in each chromosome; the orange part is the SNPs with gene information, and the blue part is the SNPs without gene information. There are some SNPs with gene information which have impacts on colon cancer and other diseases; they are shown in Table 4.8. All diseases detail were obtained from dbSNP of National Center for Biotechnology Information (NCBI), SNPedia [11] and Ensembl. The whole SNPs dataset obtained through feature selection is given in Appendix A.2.

Table 4.8: The Genome Information of our Final Results.

SNP Name	Chromosome	Gene Information	Associated Phenotype
rs17206779	5	ADAMTS6	Osteosarcoma
rs16892766	8	EIF3H	Colorectal cancer
rs7818382	8	NDUFAF6	Alzheimer's disease (late onset)
rs10795668	10	LOC105376400	Colorectal cancer
rs713065	11	PRSS23 FZD4	Familial exudative vitreoretinopathy
rs878960	15	GABRB3	autism spectrum disorder
rs1356410	15	PLA2G4F	Chagas cardiomyopathy in Tripanosoma
rs4464148	18	SMAD7	Colorectal cancer
rs12953717	18	SMAD7	Colorectal cancer
rs10411210	19	RHPN2	Colorectal cancer

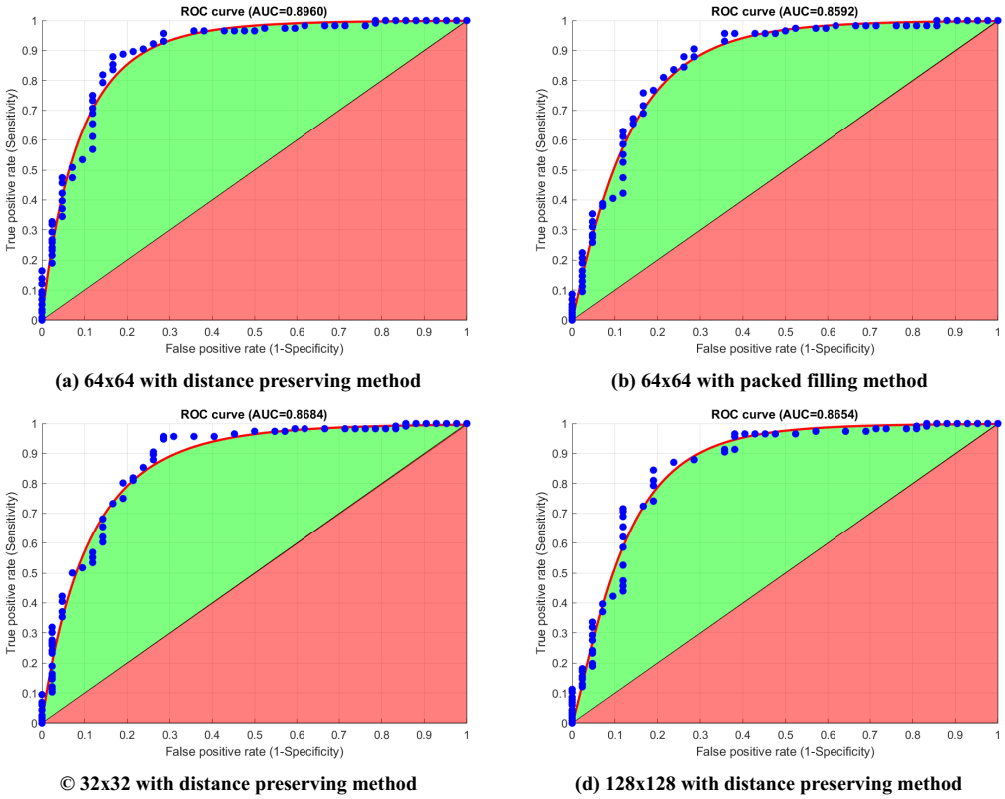

Figure 4.4: The comparison results of ROC and AUC under diferent condition. (a) is the ROC and AUC based on Hilbert curve with 64×64 size and distance preserving methods, (b) is the ROC and AUC based on Hilbert curve with 64×64 size and packed filling methods, (c) is the ROC and AUC based on Hilbert curve with 32×32 size and distance preservi methods, and (d) is the ROC and AUC based on the Hilbert curve with 128×128 size and distance preserving methods. The blue points refer to all TPR and FPR values under different conditions.

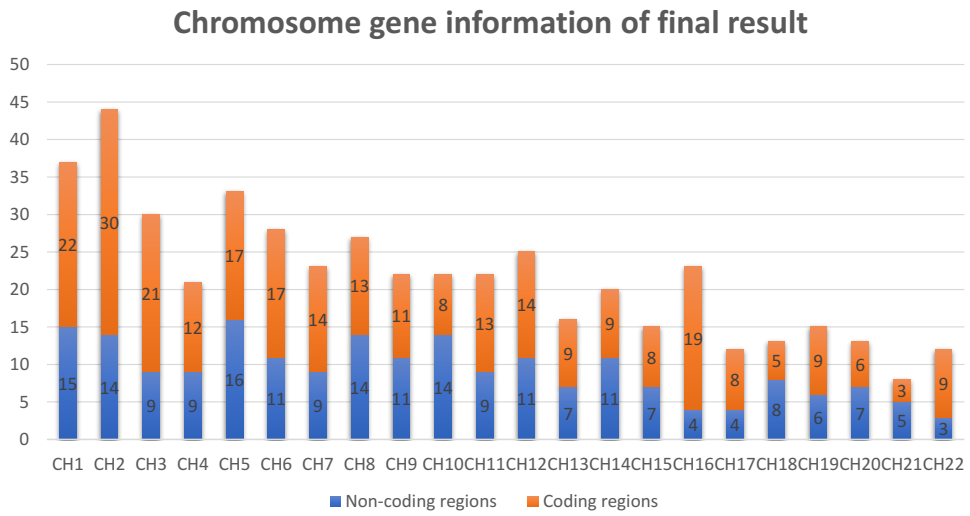

Figure 4.5: The bioinformatics details of the selected SNPs. In the same chromosome, the orange parts refer to the SNPs with gene information, the blue parts refers to the SNPs without gene information.

Chapter 5

Discussion and conclusions

5.1 Discussion

SNPs feature selection is designed to find the set of SNPs associated with a given disease. The difficulties of this study can be described from two aspects. One of the difficulties is that the quantity of the case samples is far less than the number of SNPs. For SNPs dataset with high-dimensional and small samples, how to effectively execute the mass data association analysis is a very difficult problem [71]. The other is the pathological complexity of genetic diseases, which is usually caused by multiple genes working together [71].

For the characteristics of SNP data with high-dimensional small samples, we proposes a hybrid feature selection method that effectively combines the filter method, wrapper method and deep learning. The filter method can significantly reduce the number of features for achieving the balance between the number of samples and number of features [71]. A low-dimensional SNPs dataset is therefore obtained after

extraneous features are removed. We use the wrapper method to solve the pathological complexity problem because the pathogenesis of colon cancer is not determined by a single SNP, but by some SNPs that are mutated and interact [71]. Therefore, we consider the effect of the SNPs combinations rather than the single SNP [71]. In our research, the following steps are performed: First, ReliefF is used to compute the weights for each SNP and to reserve SNP feature sets with the top highest weights (specified quantity such as 1024) as the output dataset. Then, the GA is used to process those SNPs with the highest weights. During GA processing, plenty of subsets were generated, and space-filling curve methods were used to encode these gene subsets into images. Next, the CNN is used as the classification to process these images, get the accuracies for each SNP subset and return these accuracies to GA. Finally, the SNPs subset with the best accuracy is output as the feature selection result.

The hybrid method was used to select a total of 481 SNPs as the result of feature selection in this research. This result was generated using Hilbert Curve with distance preserving filling methods in size 64×64. The classification accuracy of the selected SNPs was 90.05% based on CNN. Comparing with previous studies, we found that the amount of the final feature selection was not small enough (for example, 63 or 59 SNPs were selected as the result set [72]) in our research because the data of the classification is an image. This characteristic may result in the inclusion of some SNPs locate of outside genes in the final SNPs which were obtained through feature selection [20]. Moreover, we have not been able to determine whether these SNPs cause colon cancer without further using medical and biological experimental analysis. Generally, the feature selection result set is considered as effective if the permutation text (PT) has $PT < 0.05$ [23]; the permutation test value was 0.026 in

this project. Therefore, the feature selection set was effective and useful [23].

Each SNP includes several different attributes, such as allele and genomic distance [71]. This research presents a new hybrid method for bioinformatics research: Convert SNPs sequences to images using space-filling curves, used machine learning (such as RelifF combined with GA) to implement feature selection and CNN as classifiers. Through the analysis and comparison of the experimental results, we found that the results of SNP feature selection are not only related to the allele of SNPs but also related to the genomic distance of SNPs. Also, the results of SNPs feature selection are related to the filling method of data visualization (packed filling methods and distance preserving filling method) and the space-filling curves. In traditional analysis of GWAS data, the SNPs are the sequence data which can only represent one type attributes, such as allele [82]. However, more attributes need to be considered due to pathological complexity. Thus, adding an essential attribute (based on physical distance) into the SNP data and encoding the combined data as 2D images represent the characteristics and relations of the two types of data at the same time. Based on this idea, the newest artificial intelligence methods for classification can be used, and more data attributes and the relationships between these attributes can be considered [39], so that more comprehensive and accurate results will be attained.

When encoding the SNPs, we firstly select five different space-filling curves as the model of the image and then determine the position of the elements in the sequence. The coordinate values of the input pixel of the image are sequential. The input rank follows the direction of the spatial curve as the input lattice. After determining the position of each pixel, we convert the SNPs' value of numerical encoding to the value of each pixel in a fixed proportion according to the value of each element in the sequence,

which complete the visual image corresponding to each sequence. There are two filling methods for the SNPs sequence which are processed by data encoding: distance preserving filling methods and packed filling methods. In the distance preserving filling methods, the physical distance is considered during data encoding processing. The preserved distance is proportion to the real physical distance between different SNPs. The packed filling methods, on the other hand, do not consider the physical distance during data encoding processing, and the SNPs values are filled into the image pixel one by one depending on space-filling curve coordinates.

Overall, there are a few significant observations from our results analysis. First, the average accuracy of feature selection results, which are generated based on different space-filling curves, was different. The Hilbert curve achieved the best average accuracy of all the space-filling curves compared. Furthermore, the different average accuracies of results are obtained based on the different size images even though based on the same SNP data and the same space-filling curve. There are five types of size images for data encoding in this research, but the images with 64×64 sizes had the best average accuracy compared with others size images. Also specially, the experiments indicate different filling methods have different accuracy, even though the images possess the same data value, same space-filling curve and the same size. Our experiments also suggest using distance preserving filling methods to fill pixels in the images, which considered the physical distance between SNPs on the genome, increased the prediction accuracy of trained CNN models.

The Convolutional Neural Network (CNN) is a robust learning method, especially for image classification [1]. However, the limitations of this research study are the amount and construction of the sample set. This issue is the impact of imbalanced

training data for CNN [10]. Therefore, the oversampling technology is used in our research [81]. Its operating principle is: For each category, randomly select some images to copy, rotate, change the contrast and brightness until the number of images is the same as the one with the most significant proportion [81].

5.2 Conclusions

Traditional SNP feature selection methods are all based on SNPs sequences, which are all 1D data. The final results generated by the traditional algorithms are entirely based on the values or weights of the SNPs. Disease pathology cannot be merely determined by the single attribute such as the value of SNP, and a lot of relevant information and features must be considered. In our research, a new SNP encoding method is adapted to transform 1D SNP sequence into 2D image data, and then deep learning is used for processing and analysis. Based on this idea, our research considered the values and weights of SNP as well as the relationship between the distance (based on physical distance) of each SNP.

We use data encoding to visualize SNPs sequence to 2D data (image) and implement feature selection, which is a new research direction for GWAS. Most bioinformatics cannot be expressed as a mathematical formula like physics, and cannot be represented in a logical formula like computer science, but it can be presented by the tables, graphs, networks and other intuitive forms [40]. Therefore, scientific visualization technology can be utilized in bioinformatics. This research direction extends the representation of biological information [40] and combines the newest deep learning and traditional machine learning model to process this biological data.

Based on the implementation of hybrid methods and the results analysis, the experimental data are changed from 1D to 2D, and the space-filling curve is used due to its consideration of SNP values and their genomic distances. We found the results being affected by the feature locations, the filling method, and the data encoding method. Using the deep learning model (CNN) for classification, and keeping update and improvement of the structural change for the deep learning model, newer and more useful SNPs will be continuously discovered.

In bioinformatics, by genomic mapping, it can be determined that genes are adjacent to each other in the original genome [47]. Then, through mathematical analysis techniques, space-filling curve can be used to represent the genomic data as a 2D image [4]. The results of our experiments show the following: First, from the point of view of computer vision, these genomic data can be recoded into 2D images according to mathematical models. [4]. Second, from the point of view of deep learning, these genetic data re-encoded into 2D images can be identified, classified, etc. using various neural network models. Finally, the haplotype image constructed with the Hilbert curve has the highest accuracy. Therefore, the Hilbert curve is a good representation of genetic maps and genetic information [42].

Because multiple characteristics are considered and data encoding for feature selection of bioinformatics are used in the thesis, the final result showed that various gene information could be combined with each other in GWAS. Therefore, obtaining the SNPs feature selection result is significant in this thesis. To increase accuracy and improve the speed of processing are also crucial in our work.

5.3 Future extensions

There are several ways to improve the methods used in our work in the future. The first improvement can be made by implementing multi-dimensional data encoding [12] for SNPs in GWAS [51]. Besides, we plan to try other space-filling curves to visualize SNPs sequence. In addition, this research uses Deep Convolutional Neural Networks as our classifier, and even better results can be achieved by improving the deep learning models, including modifying the neural network structure (the amount of parameters, neural nodes and layers of CNN). Furthermore, we will use some other neural networks such as Recurrent Neural Networks [83] and Deep Belief Neural Networks [56] as classifiers. In particular, we can redesign the deep learning model to implement GWAS directly [1].

Bibliography

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. TensorFlow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[2] M. A. Ali and S. Ladhake. Overview of space-filling curves and their applications in scheduling. *International Journal of Advances in Engineering & Technology*, 1(4):148–150, 2011.

[3] X. An, D. Kuang, X. Guo, Y. Zhao, and L. He. A deep learning method for classification of EEG data based on motor imagery. In *International Conference on Intelligent Computing*, pages 203–210. Springer, 2014.

[4] S. Anders. Visualization of genomic data with the Hilbert curve. *Bioinformatics*, 25(10):1231–1235, 2009.

[5] C. Angermueller, T. Pärnamaa, L. Parts, and O. Stegle. Deep learning for computational biology. *Molecular Systems Biology*, 12(7):878, 2016.

[6] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

[7] R. Atienza. GAN by example using Keras on TensorFlow backend. *https://www.sharelatex.com/project/5a5480362f2da35852c7f43a*, pages Online; accessed 9–January–2018, 2017.

[8] A. S. A. Aziz, A. T. Azar, M. A. Salama, A. E. Hassanien, and S. E.-O. Hanafy. Genetic algorithm with different feature selection techniques for anomaly detectors generation. In *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on*, pages 769–774. IEEE, 2013.

[9] J. Brownlee. A gentle introduction to the Rectified Linear Unit(ReLU) for deep learning neural networks. *Better Deep Learning*, 9, 2019.

[10] M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *ArXiv Preprint ArXiv:1710.05381*, 2017.

[11] M. Cariaso and G. Lennon. Snpedia: A wiki supporting personal genome annotation, interpretation and analysis. *Nucleic Acids Research*, 40(D1):D1308–D1312, 2011.

[12] D. R. Chalice. A characterization of the Cantor function. *The American Mathematical Monthly*, 98(3):255–258, 1991.

[13] C. L. Chen, A. Mahjoubfar, L.-C. Tai, I. K. Blaby, A. Huang, K. R. Niazi, and B. Jalali. Deep learning in label-free cell classification. *Scientific Reports*, 6:21471, 2016.

[14] H. Chen, R. H. Chiang, and V. C. Storey. Business intelligence and analytics: from big data to big impact. *MIS Quarterly*, pages 1165–1188, 2012.

[15] Y. Chen, Y. Li, R. Narayan, A. Subramanian, and X. Xie. Gene expression inference with deep learning. *Bioinformatics*, 32(12):1832–1839, 2016.

[16] J. Davis and M. Goadrich. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 233–240. ACM, 2006.

[17] A. De La Fuente, N. Bing, I. Hoeschele, and P. Mendes. Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20(18):3565–3574, 2004.

[18] S. Diab and B. Sartawi. Classification of questions and learning outcome statements (LOS) into Blooms taxonomy (BT) by similarity measurements towards extracting of learning outcome from learning material. *arXiv preprint arXiv:1706.03191*, 2017.

[19] R. Durgabai and Y. Ravi Bhushan. Feature selection using ReliefF algorithm. *International Journal of Advanced Research in Computer and Communication Engineering*, 3(10):8215–8218, 2014.

[20] L. Elliott, K. Sharp, F. Alfaro-Almagro, G. Douaud, K. Miller, J. Marchini, and S. Smith. The genetic basis of human brain structure and function. *BioRxiv*, page 178806, 2017.

[21] R. Fakoor, F. Ladhak, A. Nazi, and M. Huber. Using deep learning to enhance cancer diagnosis and classification. In *Proceedings of the International Conference on Machine Learning*, volume 28. ACM New York, USA, 2013.

[22] O. Falola, V. C. Osamor, M. Adebiyi, and E. Adebiyi. analyzing a single nucleotide polymorphism in schizophrenia: a meta-analysis approach. *Neuropsychiatric Disease and Treatment*, 13:2243, 2017.

[23] D. François, V. Wertz, and M. Verleysen. The permutation test for feature selection by mutual information. In *ESANN*, pages 239–244, 2006.

[24] M. Friendly and D. J. Denis. Milestones in the history of thematic cartography, statistical graphics, and data visualization. *URL http://www. datavis. ca/milestones*, 32:13, 2001.

[25] E. Gawehn, J. A. Hiss, and G. Schneider. Deep learning in drug discovery. *Molecular Informatics*, 35(1):3—14, 2016.

[26] S. S. Girija. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *Software available from Tensorflow. org*, 2016.

[27] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[28] E. Y. Gorodov and V. V. Gubarev. Analytical review of data visualization methods in application to big data. *Journal of Electrical and Computer Engineering*, 2013:22–24, 2013.

[29] Z. Gu, R. Eils, and M. Schlesner. Hilbertcurve: An R/Bioconductor package for high-resolution visualization of genomic data. *Bioinformatics*, 32(15):2372–2374, 2016.

[30] R. Guerra and Z. Yu. Single nucleotide polymorphisms and their applications. In *Computational and Statistical Approaches to Genomics*, pages 311–349. Springer, 2006.

[31] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.

[32] B. M. Herrera, S. Keildson, and C. M. Lindgren. Genetics and epigenetics of obesity. *Maturitas*, 69(1):41–49, 2011.

[33] R. Herrero and V. K. Ingle. Space-filling curves applied to compression of ultraspectral images. *Signal, Image and Video Processing*, 9(6):1249–1257, 2015.

[34] J. Huang and C. X. Ling. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310, 2005.

[35] L.-L. Huang, J. Tang, D.-D. Sun, and B. Luo. Feature selection algorithm based on multi-label ReliefF. *Jisuanji Yingyong/ Journal of Computer Applications*, 32(10), 2012.

[36] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv Preprint ArXiv:1502.03167*, 2015.

[37] C. Jordan. Courbes continues. *Cours d'Analyse de l'École Polytechnique*, pages 587–594, 1887.

[38] M. Junker, R. Hoch, and A. Dengel. On the evaluation of document analysis components by recall, precision, and accuracy. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318)*, pages 713–716. IEEE, 1999.

[39] J. A. M. Kamarudin, A. Abdullah, and R. Sallehuddin. A review of deep learning architectures and their application. In *Asian Simulation Conference*, pages 83–94. Springer, 2017.

[40] D. R. Kelley, J. Snoek, and J. L. Rinn. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Research*, 26(7):990–999, 2016.

[41] J. Lanchantin, R. Singh, Z. Lin, and Y. Qi. Deep motif: Visualizing genomic sequence classifications. *arXiv preprint arXiv:1605.01133*, 2016.

[42] J. Lanchantin, R. Singh, B. Wang, and Y. Qi. Deep motif dashboard: Visualizing and understanding genomic sequences using deep neural networks. In *Pacific Symposium On Biocoputing 2017*, pages 254–265. World Scientific, 2017.

[43] C. C. Laurie, K. F. Doheny, D. B. Mirel, E. W. Pugh, L. J. Bierut, T. Bhangale, F. Boehm, N. E. Caporaso, M. C. Cornelis, H. J. Edenberg, et al. Quality control and quality assurance in genotypic data for genome-wide association studies. *Genetic Epidemiology*, 34(6):591–602, 2010.

[44] J. K. Lawder. *The application of space-filling curves to the storage and retrieval of multi-dimensional data.* PhD thesis, 2000.

[45] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, and A. Nowe. A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 9(4):1106–1119, 2012.

[46] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[47] D. Leforestier, E. Ravon, H. Muranty, A. Cornille, C. Lemaire, T. Giraud, C.-E. Durel, and A. Branca. Genomic basis of the differences between cider and dessert apple varieties. *Evolutionary Applications*, 8(7):650–661, 2015.

[48] M. K. Leung, A. Delong, B. Alipanahi, and B. J. Frey. Machine learning in genomic medicine: A review of computational problems and data sets. *Proceedings of the IEEE*, 104(1):176–197, 2016.

[49] J. Li, Y.-C. Pan, Y.-X. Li, and T.-L. Shi. Analysis and application of SNP and haplotype in the human genome. *Yi Chuan Xue Bao*, 32(8):879–889, 2005.

[50] R.-H. Li, J. X. Yu, and J. Liu. Link prediction: the power of maximal entropy random walk. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 1147–1156. ACM, 2011.

[51] M. W. Libbrecht and W. S. Noble. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16(6):321, 2015.

[52] M. Mahmud, M. S. Kaiser, A. Hussain, and S. Vassanelli. Applications of deep learning and reinforcement learning to biological data. *IEEE Transactions on Neural Networks and Learning Systems*, 29(6):2063–2079, 2018.

[53] P. Mamoshina, A. Vieira, E. Putin, and A. Zhavoronkov. Applications of deep learning in biomedicine. *Molecular Pharmaceutics*, 13(5):1445–1454, 2016.

[54] M. I. McCarthy, G. R. Abecasis, L. R. Cardon, D. B. Goldstein, J. Little, J. P. Ioannidis, and J. N. Hirschhorn. Genome-wide association studies for complex traits: consensus, uncertainty and challenges. *Nature Reviews Genetics*, 9(5):356, 2008.

[55] S. Min, B. Lee, and S. Yoon. Deep learning in bioinformatics. *Briefings in Bioinformatics*, 18(5):851–869, 2017.

[56] A.-r. Mohamed, G. E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):14–22, 2011.

[57] C. Muelder and K.-L. Ma. Rapid graph layout using space filling curves. *IEEE Transactions on Visualization and Computer Graphics*, 14(6), 2008.

[58] L. Nahlawi. *Genetic feature selection using dimensionality reduction approaches: A comparative study.* PhD thesis.

[59] C. Park, J.-Y. Koo, P. T. Kim, and J. W. Lee. Stepwise feature selection using generalized logistic loss. *Computational Statistics & Data Analysis*, 52(7):3709–3718, 2008.

[60] T. R. Patil, S. Sherekar, et al. Performance analysis of Naive Bayes and j48 classification algorithm for data classification. *International Journal of Computer Science and Applications*, 6(2):256–261, 2013.

[61] T. M. Phuong, Z. Lin, and R. B. Altman. Choosing snps using feature selection. In *Computational Systems Bioinformatics Conference, 2005. Proceedings. 2005 IEEE*, pages 301–309. IEEE, 2005.

[62] M. Pongpanich, P. F. Sullivan, and J.-Y. Tzeng. A quality control algorithm for filtering SNPs in genome-wide association studies. *Bioinformatics*, 26(14):1731–1737, 2010.

[63] S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. De Bakker, M. J. Daly, et al. PLINK: A tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics*, 81(3):559–575, 2007.

[64] L. Rampasek and A. Goldenberg. TensorFlow: Biology's gateway to deep learning? *Cell Systems*, 2(1):12–14, 2016.

[65] D. Richardson, N. Castree, M. F. Goodchild, A. L. Kobayashi, W. Liu, and R. A. Marston. *The International Encyclopedia of Geography: People, the Earth, Environment, and Technology, Volume 9.*

[66] C. A. Rietveld, S. E. Medland, J. Derringer, J. Yang, T. Esko, N. W. Martin, H.-J. Westra, K. Shakhbazov, A. Abdellaoui, A. Agrawal, et al. GWAS of 126,559 individuals identifies genetic variants associated with educational attainment. *Science*, 340(6139):1467–1471, 2013.

[67] M. Robnik-Šikonja and I. Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, 53(1-2):23–69, 2003.

[68] S. Saha. A comprehensive guide to convolutional neural networks. *Towards Data Science*.

[69] J. Salamon and J. P. Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, 2017.

[70] N. J. Schork, D. Fallin, and J. S. Lanchbury. Single nucleotide polymorphisms and the future of genetic epidemiology. *Clinical Genetics*, 58(4):250–264, 2000.

[71] F. R. Schumacher, S. L. Schmit, S. Jiao, C. K. Edlund, H. Wang, B. Zhang, L. Hsu, S.-C. Huang, C. P. Fischer, J. F. Harju, et al. Genome-wide association study of colorectal cancer identifies six new susceptibility loci. *Nature Communications*, 6:7138, 2015.

[72] S. C. Shah and A. Kusiak. Data mining and genetic algorithm based gene/snp selection. *Artificial Intelligence in Medicine*, 31(3):183–196, 2004.

[73] S. Sudholt and G. A. Fink. Evaluating word string embeddings and loss functions for CNN-based word spotting. In *2017 14th IAPR International Conference on*

Document Analysis and Recognition (ICDAR), volume 1, pages 493–498. IEEE, 2017.

[74] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017.

[75] S. Uppu, A. Krishna, and R. P. Gopalan. Towards deep learning in genome-wide association interaction studies. In *PACIS*, page 20, 2016.

[76] R. Upstill-Goddard, D. Eccles, J. Fliege, and A. Collins. Machine learning approaches for the discovery of gene–gene interactions in disease data. *Briefings in Bioinformatics*, 14(2):251–260, 2012.

[77] P. Urysohn. Works on topology and other areas of mathematics 1, 2. *State Publ. of Technical and Theoretical Literature, Moscow*, 1951.

[78] D. Ververidis and C. Kotropoulos. Fast and accurate sequential floating forward feature selection with the Bayes classifier applied to speech emotion recognition. *Signal Processing*, 88(12):2956–2970, 2008.

[79] M. O. Ward, G. Grinstein, and D. Keim. *Interactive Data Visualization: Foundations, Techniques, and Applications*. AK Peters/CRC Press, 2015.

[80] D. Welter, J. MacArthur, J. Morales, T. Burdett, P. Hall, H. Junkins, A. Klemm, P. Flicek, T. Manolio, L. Hindorff, et al. The NHGRI GWAS catalog, a curated resource of SNP-trait associations. *Nucleic Acids Research*, 42(D1):D1001–D1006, 2013.

[81] Y. Yan, M. Chen, M.-L. Shyu, and S.-C. Chen. Deep learning for imbalanced multimedia data classification. In *Multimedia (ISM), 2015 IEEE International Symposium on*, pages 483–488. IEEE, 2015.

[82] J.-B. Yang, K.-Q. Shen, C.-J. Ong, and X.-P. Li. Feature selection for MLP neural network: The use of random permutation of probabilistic outputs. *IEEE Transactions on Neural Networks*, 20(12):1911–1922, 2009.

[83] W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. *ArXiv Preprint ArXiv:1409.2329*, 2014.

[84] H. Zeng, M. D. Edwards, G. Liu, and D. K. Gifford. Convolutional neural network architectures for predicting dna–protein binding. *Bioinformatics*, 32(12):i121–i127, 2016.

[85] G. Zhang, R. J. Doviak, J. Vivekanandan, W. O. Brown, and S. A. Cohn. Cross-correlation ratio method to estimate cross-beam wind and comparison with a full correlation analysis. *Radio Science*, 38(3):17–1, 2003.

[86] J. Zhou and O. G. Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature Methods*, 12(10):931, 2015.

[87] C. Zhu. Review of dropout: A simple way to prevent neural networks from overfitting. *Towards Data Science*.

[88] M. H. Zweig and G. Campbell. Receiver-operating characteristic (ROC) plots: A fundamental evaluation tool in clinical medicine. *Clinical Chemistry*, 39(4):561–577, 1993.

Appendix A

Appendices

A.1 Python Code of Discriminator

See the website: https://github.com/youraustin/GWAS-CNN-/tree/master/Tensorflow

A.2 The Genome Information of Final Results

See the website: https://github.com/youraustin/GWAS-CNN-/blob/master/The%20Genome%20information%20of%20final%20results.txt